

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«На правах рукопису»
УДК _____

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 2021 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-науковою програмою «Інтегровані інформаційні системи»
зі спеціальності 126 «Інформаційні системи та технології»**

на тему: «Моделювання динамічних процесів крокуючого робота»

Виконав:
студент VI курсу, групи ІА-91мн
Сверчков Владислав Андрійович _____

Керівник:
доцент каф. АУТС, к.т.н., доцент
Репнікова Наталія Борисівна _____

Консультант з _____:

Рецензент:
доцент каф. АСОІУ, к.т.н., доцент
Жданова Олена Григорівна _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-наукова програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 2021 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Сверчкову Владиславу Андрійовичу

1. Тема дисертації «Моделювання динамічних процесів крокуючого робота», науковий керівник дисертації Репнікова Наталія Борисівна, к.т.н., доц., затверджені наказом по університету від «12» березня 2021 р. № 809-с
2. Термін подання студентом дисертації 11.05.2021
3. Об'єкт дослідження: модель двоногого крокуючого робота
4. Предмет дослідження: моделювання динамічних процесів крокуючого робота. Маса робота – 172 кг; зріст – 100 см; швидкість виконання роботою завдання – 0.1391 м/с; споживана потужність протягом виконання завдання – 13.999 кВт; час виконання завдання – 22.11 с; можливість обходити перешкоду з подальшим відновленням початкового прямолінійного руху
5. Перелік завдань, які потрібно розробити: огляд та аналіз існуючих рішень; опис математичної моделі антропоморфного механізму; розроблення структурної схеми модернізованої моделі двоногого робота; розроблення вдосконаленої моделі робота у середовищі MATLAB/Simulink; опис запропонованих модернізацій робота; опис концепції обходження перешкоди роботою; програмування визначення середньої швидкості моделі робота протягом усього часу моделювання; програмування визначення динамічних процесів та споживаної потужності моделі робота; обчислення CoT робота та знаходження оптимального значення.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: математична модель двоногого крокуючого механізму; структурна схема моделі робота Darth EvadeR; блок-схема алгоритму поведінки робота; модель робота Darth EvadeR; графік пройденої роботою відстані; графіки залежності кутової швидкості, обертального моменту, споживаної потужності та станів робота від часу моделювання для суглобу стегна правої кінцівки; графік швидкості робота Darth

EvadeR; відпрацювання коду для визначення потужності складових частин робота; порівняння CoT моделі робота Darth EvadeR з іншими роботами.

7. Орієнтовний перелік публікацій: стаття «Вдосконалення функціональних можливостей моделі крокуючого робота», тези «Дослідження ефективності моделі двоногого крокуючого робота»

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 01.02.2021

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Отримання завдання та узгодження вихідних даних	01.02.2021	
2.	Огляд та аналіз існуючих рішень	02.02.2021-07.02.2021	
3.	Опис математичної моделі антропоморфного механізму	08.02.2021-13.02.2021	
4.	Розроблення структурної схеми модернізованої моделі двоногого робота	14.02.2021-16.02.2021	
5.	Розроблення вдосконаленої моделі робота у MATLAB/Simulink	17.02.2021-15.03.2021	
6.	Опис запропонованих модернізацій робота	16.03.2021-20.03.2021	
7.	Опис концепції обходження перешкоди роботом	21.03.2021-23.03.2021	
8.	Програмування визначення середньої швидкості моделі робота протягом усього часу моделювання	24.03.2021-01.04.2021	
9.	Програмування визначення динамічних процесів та споживаної потужності моделі робота	02.04.2021-06.04.2021	
11.	Обчислення CoT робота та знаходження оптимального значення	07.04.2021-19.04.2021	
12.	Оформлення дисертації	20.04.2021-04.05.2021	
13.	Подання дисертації до попереднього захисту	05.05.2021	

Студент

Владислав СВЕРЧКОВ

Науковий керівник

Наталія РЕПНІКОВА

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Магістерська дисертація на здобуття ступеня «магістр» за освітньо-науковою програмою підготовки «Інтегровані інформаційні системи» на тему «Моделювання динамічних процесів крокуючого робота». Дисертація містить 103 сторінки, 49 рисунків, 7 таблиць, 3 додатки, 46 джерел.

Актуальність. Сучасний розвиток науки дозволяє проводити активні дослідження в області робототехніки з метою створення спеціалізованих двоногих роботів, котрі будуть асистувати людині у небезпечних середовищах або навіть повністю замінювати її. Тому задача розширення функціональних можливостей двоногих роботів для використання у небезпечних середовищах є досить цікавою, а її розв'язок – затребуваним.

Метою магістерської дисертації є підвищення функціональних можливостей існуючої моделі двоногого крокуючого робота.

Об'єкт дослідження: модель двоногого крокуючого робота у середовищі MATLAB/Simulink.

Предмет дослідження: моделювання динамічних процесів крокуючого робота.

Наукова новизна одержаних у магістерській дисертації результатів полягає у вдосконаленні функціональних можливостей існуючої моделі двоногого крокуючого робота, а саме – у реалізації логіки обходження моделлю робота завади на її шляху.

Публікація результатів дисертації. За результатами роботи було опубліковано статтю:

Репнікова Н. Б. Вдосконалення функціональних можливостей моделі крокуючого робота / Н. Б. Репнікова, В. А. Сверчков // Проблеми інформатизації та управління: збірник наукових праць. – 2021. — Т. 1, №65 — С. 62-67.
<https://doi.org/10.18372/2073-4751.65.15372>

Ключові слова: ДВОНОГИЙ КРОКУЮЧИЙ РОБОТ, ЕФЕКТИВНІСТЬ РОБОТА, СОТ, ОБХОДЖЕННЯ ПЕРЕШКОД, MATLAB/SIMULINK.

ABSTRACT

Master's dissertation on the educational-scientific level training program “Master” on the theme "Modeling of dynamic processes of a walking robot". The dissertation contains 103 pages, 49 figures, 7 tables, 3 applications, 46 sources.

Relevance. Modern level of science brings huge possibilities for active research in the field of robotics in order to create specialized bipedal robots that will assist human in dangerous environments or even completely replace him or her. Therefore, the task of expanding the functionality of bipedal robots for use in hazardous environments is quite interesting and topical.

The purpose of the master's dissertation is to increase the functionality of the existing model of a two-legged walking robot.

Object of research: model of a bipedal walking robot in MATLAB/Simulink environment.

Subject of research: modeling of dynamic processes of a walking robot.

The scientific novelty of the results obtained in the master's dissertation is improvement of functionality of existing two-legged walking robot's model, namely - to implement the logic of the obstacle bypassing by the robot's model on its path.

Publication of dissertation's results. According to the results of the work, an article was published:

Репнікова Н. Б. Вдосконалення функціональних можливостей моделі крокуючого робота / Н. Б. Репнікова, В. А. Сверчков // Проблеми інформатизації та управління: збірник наукових праць. – 2021. — Т. 1, №65 — С. 62-67.
<https://doi.org/10.18372/2073-4751.65.15372>

Keywords: TWO-LEGGED WALKING ROBOT, EFFICIENCY OF ROBOT, COT, OBSTACLE AVOIDANCE, MATLAB/SIMULINK.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	11
1.1 Загальна характеристика двоногих крокуючих роботів та їх історія	11
1.2 Гуманоїдний робот ASIMO.....	14
1.3 Гуманоїдний робот DURUS	17
1.4 Рятувальник-акробат Atlas	19
1.5 Робот-вогнеборець SAFFiR.....	21
1.6 Робот TORO	23
1.7 Порівняльний аналіз розглянутих ДКР	25
1.8 Спільна проблематика розглянутих ДКР	27
Висновки до розділу 1	30
2 РОЗРОБЛЕННЯ ВДОСКОНАЛЕНОЇ МОДЕЛІ РОБОТА У СЕРЕДОВИЩІ MATLAB/SIMULINK	31
2.1 Опис математичної моделі антропоморфного механізму.....	31
2.2 Розроблення структурної схеми авторської моделі двоногого робота, виконаної у MATLAB/Simulink	39
2.3 Структурна схема модернізованої моделі двоногого крокуючого робота, виконаної у MATLAB/Simulink	41
2.4 Будова вдосконаленої моделі робота DARTH EvadeR у середовищі MATLAB/Simulink	44
Висновки до розділу 2	52
3 ОПИС ЗАПРОПОНОВАНИХ МОДЕРНІЗАЦІЙ МОДЕЛІ РОБОТА DARTH EVADER	54

3.1 Характеристики авторського двоногого робота	54
3.2 Шляхи та результати модернізації моделі Darth EvadeR.....	55
3.3 Опис концепції обходження перешкоди роботом	57
3.4 Розроблення блок-схеми алгоритму функціонування моделі робота	63
3.5 Реалізація за допомогою засобів Simulink логіки переміщення робота при виявленні перешкоди	64
3.6 Програмування визначення середньої швидкості моделі робота протягом усього часу моделювання.....	68
3.7 Формула для визначення середньої потужності, котру споживає модель робота	70
3.8 Програмування визначення динамічних процесів та споживаної потужності моделі робота Darth EvadeR	72
Висновки до розділу 3	79
4 АНАЛІЗ ТА ОБРАХУВАННЯ ЕФЕКТИВНОСТІ МОДЕЛІ КРОКУЮЧОГО РОБОТА DARTH EVADER	81
4.1 Загальні теоретичні відомості про ефективність переміщення живих істот та штучних механізмів	81
4.2 Обчислення CoT робота та знаходження оптимального значення.....	84
4.3 Порівняння отриманого оптимального CoT з існуючими роботами.....	92
Висновки до розділу 4	95
ВИСНОВКИ.....	96
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	98
ДОДАТОК А Код обчислення швидкості моделі робота	104
ДОДАТОК Б Код для обчислення потужності моделі робота і відстежування динамічних процесів	105
ДОДАТОК В Перелік опублікованих матеріалів	123

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ДКР — двоногий крокуючий робот

КР — крокуючий робот

ШІ — штучний інтелект

ПЗ – програмне забезпечення

ВСТУП

Двадцять перше століття неможливо уявити без робототехнічних систем. Автономні роботи-маніпулятори для збору врожаю, компонування електронних мікросхем або збору нового автомобіля, активні екзоскелети, що допомагають людям з вадами нижніх кінцівок у реабілітації, квадрокоптери, що здатні знімати чудові панорамні відео з висоти пташиного польоту, а також моніторити ворожі позиції під час військових дій — це лише вершина айсбергу реалій сьогодення.

Серед усіх робототехнічних досягнень виділяються двоногі крокуючі роботи (ДКР), які мають дві кінцівки і при цьому є антропоморфними, тобто такими, що наближено відтворюють будову тіла людини (або лише її комплекс нижніх кінцівок). ДКР кидають виклик науковцям, оскільки дослідження та моделювання систем подібного роду ще не досягло свого апогею і не змогло знайти універсального рішення, яке б дозволило запустити масове виробництво подібних роботів. Це дозволяє вченим щоразу розкривати нові особливості та тонкощі функціонування кінцівок, що виливається у розробку більш ефективних крокуючих роботів (КР).

Актуальність досліджень у напрямку ДКР підтверджується затребуваністю робототехнічних систем, котрі могли б виконувати певні транспортні задачі, допомагати людям у небезпечних середовищах (пожежі, хімічно та радіаційно забруднені території, зони проведення військових дій), чи навіть особам з частковою або повною паралізованістю ніг в реабілітаційних заходах [1, 2]. Особлива потреба у роботах, котрі могли б розпізнавати перешкоди та уникати їх, оскільки небезпечні середовища мають не незайману поверхню, а фактично являють собою суцільну полосу перешкод.

В природі існує багато натуральних ландшафтів з нерівною поверхнею: пагорби, гори, різноманітні узбережжя і т. д. Вони створюють незручні умови для переміщення роботів з колесами або гусеничними стрічками. Це також підвищує актуальність ходячих роботів, які відтворюють рухи певних представників фауни, що розповсюджена у конкретній місцевості [3]. Важливо розуміти, що людина —

істота, яка може вдало переміщуватися як лісистими та кам'янистими пагорбами, так і піщаними дюнами. Тому задача розроблення більш ефективних і функціональних ДКР є перспективною та затребуваною.

Мета: підвищення функціональних можливостей існуючої моделі двоногого крокуючого робота.

Для досягнення поставленої мети вирішені наступні **задачі:**

- огляд та аналіз існуючих рішень двоногих крокуючих роботів та їх функціональних можливостей;
- зміна ваги моделі робота на більш наближену до людської;
- зміна розмірів складових частин моделі робота на такі, що наближені до людського комплексу нижніх кінцівок;
- розширення кількості степенів свободи моделі робота з 6 до 12;
- вдосконалення способу ходьби моделі робота з огляду на збільшену масу та габарити;
- реалізація детектування моделлю робота перешкоди на шляху;
- розроблення та реалізація алгоритму обходження моделлю робота перешкоди з наступним відновленням початкової прямолінійної ходьби;
- оптимізація швидкості та часу зближення вдосконаленої моделі робота з перешкодою та її обходження;
- дослідження ефективності вдосконаленої моделі робота;
- дослідження впливу тривалості циклу ходьби на швидкість відпрацювання моделі робота.

Для вирішення задач магістерської дисертації використані наступні **методи:**

- метод комп'ютерного моделювання;
- фізико-математичні формули;
- алгоритми, конструкції та моделі Matlab/Simulink;
- метод аналізу ефективності.

Об'єктом дослідження є модель двоногого крокуючого робота, а **предметом** — моделювання динамічних процесів крокуючого робота.

Наукова новизна одержаних у магістерській дисертації результатів полягає у вдосконаленні функціональних можливостей існуючої моделі двоногого крокуючого робота, а саме – у реалізації логіки обходження моделлю робота завади на її шляху.

Публікація результатів дисертації. Відповідно до результатів роботи опубліковано статтю:

Репнікова Н. Б. Вдосконалення функціональних можливостей моделі крокуючого робота / Н. Б. Репнікова, В. А. Сверчков // Проблеми інформатизації та управління: збірник наукових праць. – 2021. — Т. 1, №65 — С. 62-67.
<https://doi.org/10.18372/2073-4751.65.15372>.

Апробація. Результати досліджень магістерської дисертації обговорювались на XI Міжнародній науково-практичній конференції Summer InfoCom Advanced Solutions 2021, яка відбулася 13 та 14 травня 2021 року; при цьому опубліковані тези на тему «Дослідження ефективності моделі двоногого крокуючого робота».

1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Загальна характеристика двоногих крокуючих роботів та їх історія

Двоногий крокуючий робот — це підвид крокуючих роботів, який переміщується на двох кінцівках за допомогою ходьби з підтриманням балансу [1]. Хоча дане визначення не дає ствердження, що ДКР — це обов'язково людиноподібний робот, переважна кількість роботів даного виду наслідує саме людську морфологію.

Монографія [4] визначає двоногого робота як кінематичний ланцюг, що складається з двох субланцюгів, які називаються ногами, а також (зазвичай) ще один субланцюг, який називається торсом (тулубом), причому усі субланцюги з'єднані спільною точкою — стегном. Обидві ноги можуть контактувати з опорною поверхнею через кінці ніг, котрі зазвичай мають додаткові данки, що звуться стопами. Коли тільки одна нога знаходиться в контакті з опорною поверхнею, ця нога називається опорною, в той час, як друга — переносною.

Зацікавленість у побудові антропоморфних механізмів, котрі можуть відтворювати рухи людини, сягає глибокої давнини. Ще в далекому XIV-му столітті великий мислитель епохи Відродження — Леонардо да Вінчі — спроектував (а, можливо, навіть і побудував) перший в історії західної цивілізації людиноподібний робот у вигляді середньовічного лицаря, закутого в лати, котрий за задумкою міг би відтворювати деякі рухи людини, як-от: махи руками, рухи головою, щелепою тощо [5].

Дослідження локомоції ДКР (пересування ДКР у просторі з використанням кінцівок) були повільнішими за дослідження роботів з більшою кількістю ніг через головну проблему роботів даного типу — необхідність наявності стабільного контролю через високі вимоги до динамічного балансу. Робот мав відтворювати ефективну та стійку ходьбу без падіння через втрату балансу. До XX-го сторіччя це сильно заважало науковцям, але з розвитком електроніки та появою комп'ютерних

систем дослідження поновилися і перейшли у якісно нову фазу. Пік уваги до КР припав на 70-80-ті роки — саме тоді були створені перші двоногі роботи [6].

У 1968 році Міомір Вукобратович — сербський вчений та видатний спеціаліст в області біомеханіки та робототехніки — висунув концепцію точки нульового моменту (англ. «Zero Moment Point», або ж «ZMP»), яка часто використовується як критерій стабільності ДКР і при цьому виявилася дуже важливою в плануванні руху подібних роботів [7]. В цей же час професор Вукобратович приступає до розробки першого активного екзоскелету в інституті імені Михайла Пупіна.

Паралельно з розвитком робототехніки на заході, про свої амбіції на цьому поприщі стала заявляти Японія.

Дослідження велися покроково. Спочатку в 1966-1967 рр. на основі аналізу рухів нижніх кінцівок був спроектований та побудований ДКР WL-1, що дозволило визначити фундаментальні засади антропоморфного переміщення на двох кінцівках. У 1969-му році була представлена модель WL-3 — механічна модель нижніх кінцівок, котра мала електрогідравлічний сервопривід та функціонувала за принципом Master-Slave. Даний екземпляр демонстрував антропоморфні рухи ногами, а також міг сідати та вставати. Разом з WL-3 був розроблений робот WAP-1 із гумовими м'язами, котрий функціонував за рахунок пневматичного педипулятору [6].

Період 70-80-х років був дуже продуктивним для університету Васеда — тоді було випущено ще багато різних варіацій ДКР. Не дивлячись на велику кількість різних робототехнічних екземплярів, до світової історії даний заклад вищої освіти потрапив саме завдяки WABOT-1 (WAseda roBOT). Це двоногий крокуючий робот, розробка якого була анонсована ще у 1967-му році, а вже 1972-го було презентовано перший у світі антропоморфний робот у натуральну величину з власним інтелектом (рисунок 1.1).

WABOT-1 мав 3 основні системи: контролю кінцівок, зорову та мовну. Він вмів спілкуватися японською мовою, вимірювати відстань та напрямок до об'єктів за рахунок зовнішніх рецепторів, якими виступали штучні вуха та очі, а також захоплювати і транспортувати предмети власними руками завдяки наявним

тактильним сенсорам. Науковці університету Васеда визначили, що WABOT-1 за розумовими здібностями більше всього наближений до півторарічної дитини [8].

Його нащадок — WABOT-2, випущений у 1984-му році, відійшов від ідеї переміщень у просторі, але натомість чудово демонстрував вміння грати на клавішних інструментах, читаючи необхідні ноти через голову-камеру, або дістаючи необхідні мелодії зі своєї пам'яті [8]. Екземпляр можна побачити на рисунку 1.1.

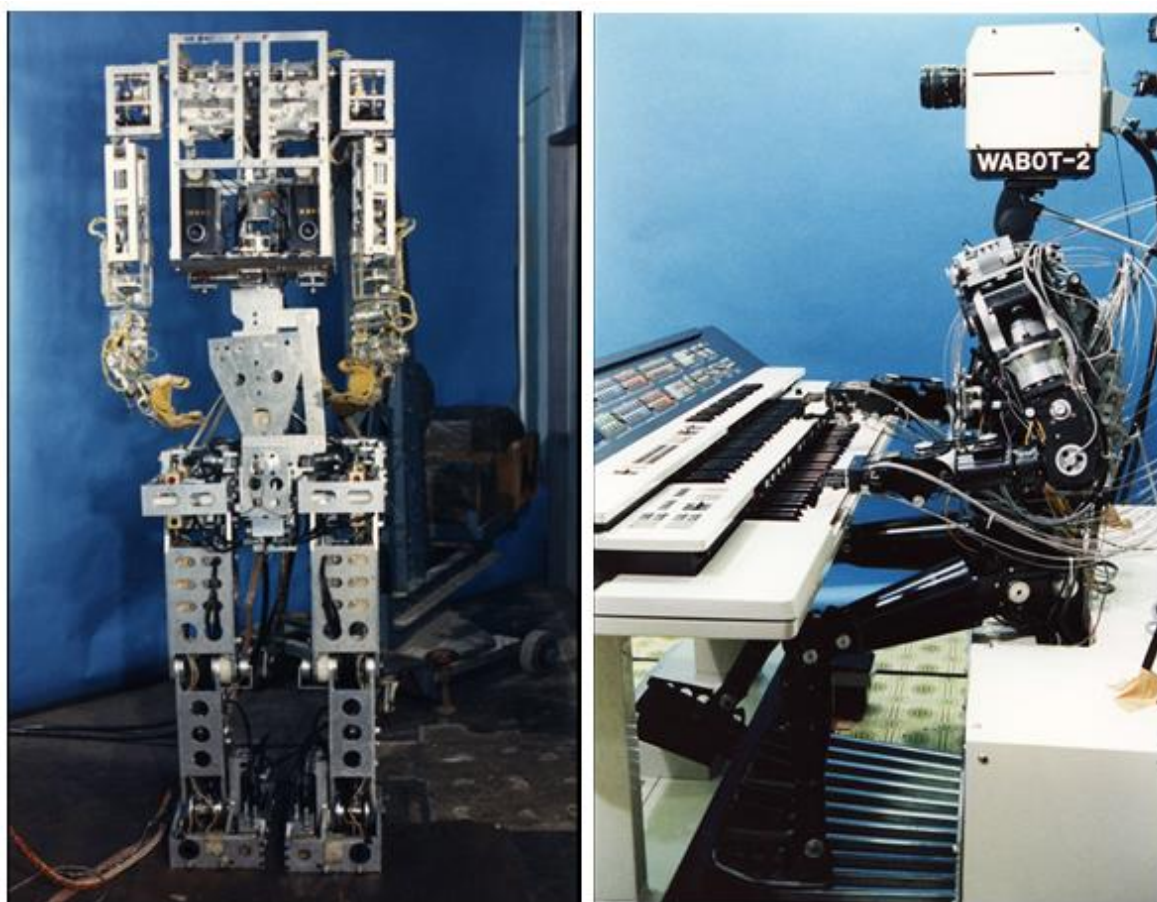


Рисунок 1.1 — WABOT-1 — перший у світі двоногий крокуючий робот у повний зріст, котрий має інтелект (зліва) [9], а також WABOT-2 — робот, котрий вміє грати на клавішних інструментах (справа) [10]

У 1989 році в Канаді Тед Макгір розробляє концепцію пасивної динамічної ходьби, котра спрямована на вдосконалення системи ДКР і при цьому передбачає використання природньої локомоції без використання контролерів, сервоприводів та

взагалі будь-якої електроніки, що також дає свої плоди. Додатково починають виникати ідеї залучення та використання нейронних мереж для різноманітних задач [11].

На початку ХХІ ст., не дивлячись на досягнення у робототехніці ДКР, досі існувала прірва між фізичними здібностями роботів та контролерами, що ними керують. Це призводило до неможливості розкриття справжнього потенціалу ДКР. Для того, щоб використати переваги пасивної динамічної ходьби та контролерів, що використовуються для стабілізації і управління роботом, Джессі Грізл на початку двохтисячних запропонував концепцію гібридної нульової динаміки (англійською «Hybrid Zero Dynamic», або ж «HZD»), котра описує можливість досягнення посправжньому динамічної ходьби та дозволяє працювати над бігом роботів та їх дійсно ефективним переміщенням. Частка «гібридний» вказує на те, що концепція HZD передбачає наявність як дискретної, так і неперервної нелінійної динаміки у роботів [12].

Зважаючи на всі успіхи у створенні ефективних ДКР, беззаперечним є те, що Японія як ніяка інша країна зробила дуже гучний старт у розробці гуманоїдних роботів і навіть на сьогоднішній день вона продовжує займати домінуючі позиції у цій царині. Наряду з нею йдуть такі країни, як США, Китай, Південна Корея та Франція. Однак, у кожної країни власні напрямки розробки. У Японії фокус йде на максимально правдоподібне відтворення людської локомоції та поведінки; європейські країни зацікавлені у розробці ДКР для медичних потреб, а американці прагнуть досягти успіху на військовому поприщі [11].

В наступних розділах будуть детально розглянуті конкретні ДКР, їх характеристики, особливості побудови та функціональні можливості.

1.2 Гуманоїдний робот ASIMO

ASIMO — це гуманоїдний робот від компанії Honda Motor Company. Назва робота походить від виразу Advanced Step in Innovative MObility. 1986-го року Honda розпочала працювати над роботами і випустила 7 роботів серії E (E0 – E6), з

роботи серії P (P1 – P3), а також розумний робот ASIMO (у 2000-му році). Його зображено на рисунку 1.2.



Рисунок 1.2 – Робот-гуманоїд Honda ASIMO [13]

Відповідно до джерела [14], ASIMO двохтисячного року відрізняється наступними властивостями:

- компактні розміри та мала вага;
- розширені технології ходьби (технологія i-WALK для передбачення наступного руху);
- просте дистанційне управління;
- широкі можливості руху за рахунок великої кількості степенів свободи;
- приємний дизайн.

Технічні характеристики робота ASIMO:

- вага: 52 кг;
- висота: 1200 мм;
- довжина: 440 мм;
- ширина: 450 мм;

- швидкість ходьби: 0...1.6 км/год;
- кількість степенів свободи: 2 для голови, 10 для рук, 2 для долонь, 12 для ніг (сумарно – 26);
- виконавчі механізми: сервомотор, гармонійний уповільнювач, блок управління двигуном;
- контролер: блок управління ходьбою та рухами робота, блок управління бездротової передачі;
- сенсори: 6-осьовий датчик для стоп, гіроскоп та датчик уповільнення для торсу;
- джерело живлення: 38.4V/10AH (нікель-гідридні батареї), 30 хв автономного живлення;
- керування: робоча станція та портативний контролер.

Більш сучасні версії ASIMO отримали розширену функціональність, яка дозволила вивести цей тип роботів на новий рівень. Серед нововведень: технічний зір, синтез та розпізнавання мови, доставка об'єктів, вміння користуватися сходами, прийняття команд від людини та їх виконання, розпізнавання об'єктів, що рухаються, поз та жестів, навколишнього середовища, людських облич.

Також була додана сумісність з інтернетом і можливість розмовляти та розпізнавати співбесідника — тепер ASIMO може вітати людей японською та англійською, відповідати на запитання про погоду, новини тощо. Нововведення у вигляді інтеграції з користувацькою мережею дозволить приймати дані, наприклад, щодо імен та зовнішності певних людей (відвідувачів публічних закладів) і т. д. Вага оновленого робота зменшилася до 48 кг, довжина до 340 мм, а зріст збільшився до 1300 мм [15].

При цьому новий ASIMO зміг розвивати швидкість до 2.7 км/год, отримав здатність бігати (до 9 км/год) з часом перебування у повітрі у розмірі 0.08 с. Нова літій-іонна батарея (51.8 В, 6 кг) дозволяє працювати в автономному режимі близько години. Кількість степенів свободи зросла до 57 (3 для голови, 14 для рук, 26 для долонь, 2 для тулуба, 12 для ніг).

Робот ASIMO має ще багато чудових та приємних функцій, котрі тут не будуть згадуватися, однак, одне важливе рішення японських розробників ми відмітимо окремо. Це спосіб уникнення роботом перешкод, котрий має максимальну наглядність на рисунку 1.3.

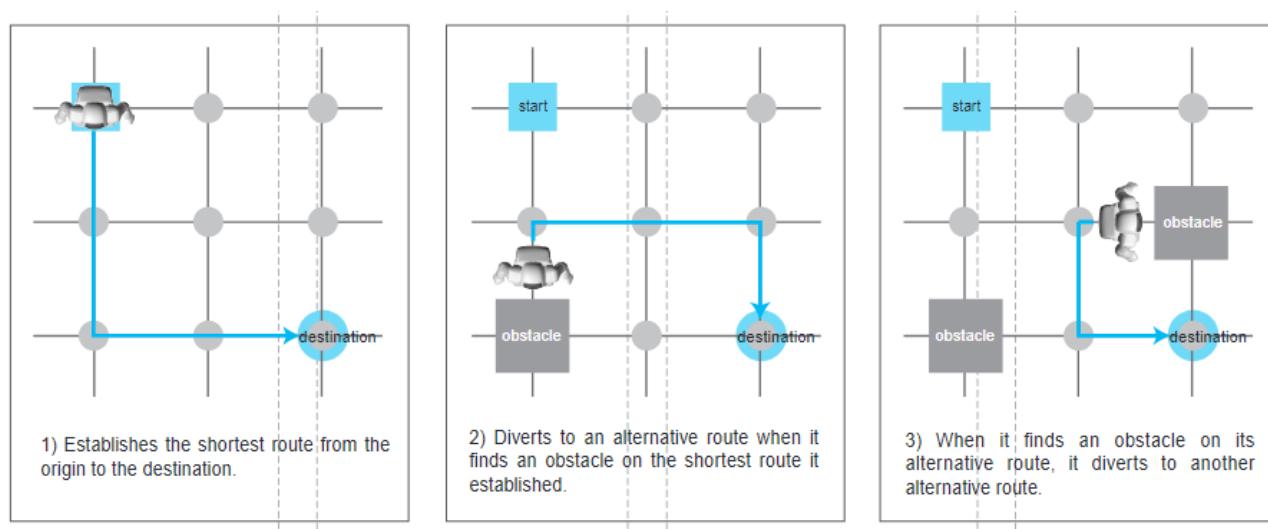


Рисунок 1.3 – Спосіб обходження перешкод роботом ASIMO [15]

Відповідно до рисунку 1.3, робот спочатку визначає найкоротший маршрут, починає рухатися, а при фіксуванні перешкоди самостійно розробляє альтернативний шлях до цілі.

1.3 Гуманоїдний робот DURUS

DURUS — це двоногий робот американських дослідників з AMBER Lab (Технологічний інститут Джорджії) та SRI Robotics, котрий позиціонується як ультра ефективний двоногий робот. Даний гуманоїд побачив світ у 2015 році; важить 100 кг, має 23 степені свободи, а також автономну батарею на 8 год потужністю 2.2 кВт/год [16].

Ключова відмінність DURUS від інших роботів-гуманоїдів полягає у використанні пасивних пружин у щиколотках, які не потребують живлення — це

дає виграш у ефективності робота, мірилом якої виступає показник CoT [17]. Про цей показник детальніше будемо говорити у розділі 4, а зараз лише скажемо, що це індекс використання енергії тіла (як живої істоти, так і робототехнічної системи) для переміщення на певну відстань. CoT DURUS`а коливається від 1.6 до 1.5, що є дуже гарним показником енергоефективності [18]. Сам робот зображений на рисунку 1.4.

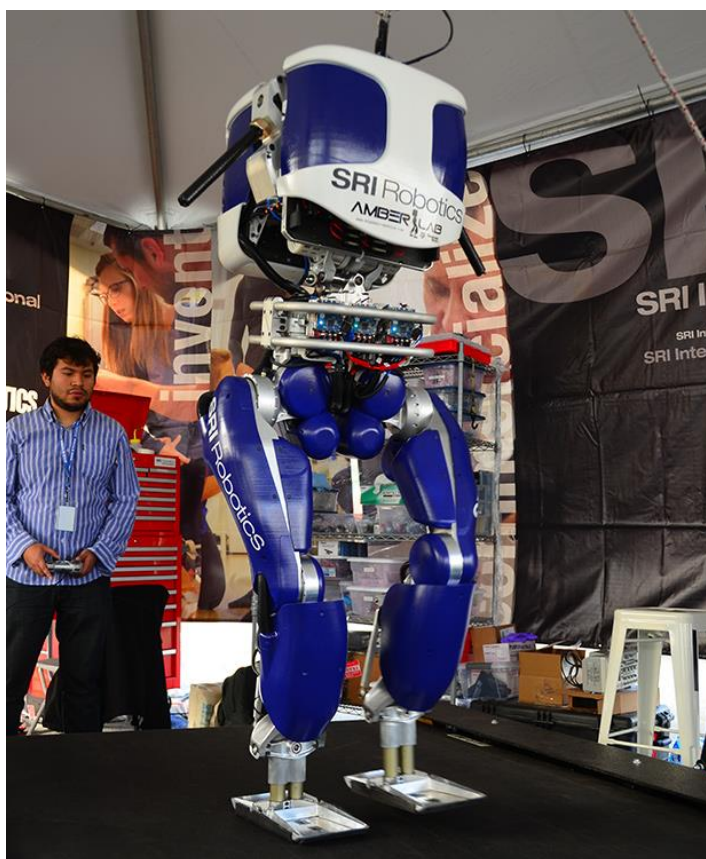


Рисунок 1.4 – Робот DURUS від AMBER Lab [18]

DURUS, використовуючи дві батареї, за 4 год 52 хв 55 с проходить відстань 3867.8 м, а отже його швидкість становить близько 0.22 м/с [17]. Робот зосереджений на реалізації ефективної прямолінійної ходьби за допомогою органічного поєднання пасивних елементів та активних приводів. Однак, окрім цієї функції гуманоїд нічого не вміє (як-от виявлення та обходження перешкод, користування сходами і т. д.). DURUS є чудовим прикладом того, якою має бути якісна з точки зору енергозатрат ходьба.

1.4 Рятувальник-акробат Atlas

Atlas — гуманоїдний робот, розроблений американськими науковцями з компанії Boston Dynamics на замовлення урядової структури DARPA. Перша версія робота з'явилася у 2013 році, але дослідження ведуться і до цього дня. Можливості Atlas`у вражають, оскільки йому під силу виконувати різноманітні акробатичні трюки, як-от: перекид, біг, стійка на руках, стрибки на високі об'єкти, перестрибування об'єктів, паркур, зворотнє сальто, стрибок та обертання у повітрі навколо своєї осі на 360 градусів з успішним приземленням на обидві кінцівки — і все це у неперервному режимі [19]. Усі паркурні навички робота можна побачити на власні очі за посиланням [20]. Сам робот зображений на рисунку 1.5.

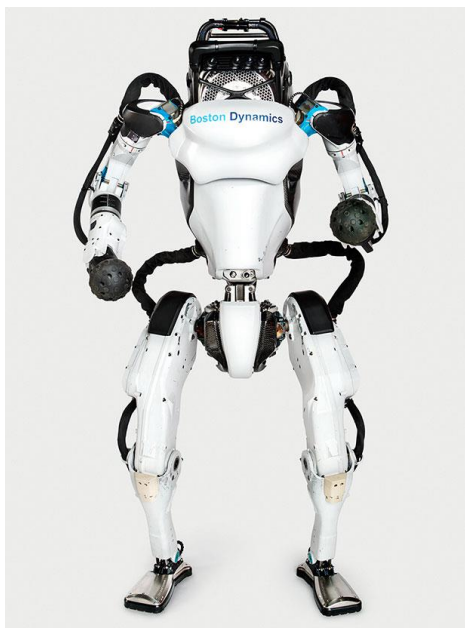


Рисунок 1.5 – Робот Atlas від Boston Dynamics [21]

Відповідно до заяв самих розробників, Atlas вважається найбільш динамічним гуманоїдним роботом, котрий створений не тільки для повторення людської локомоції, але й для розширення її можливостей за межі людських [21]. В основі такої феноменальної мобільності лежать вдосконалені алгоритми управління

роботом, що реалізують когнітивний та рухомий інтелект робота. Перший дозволяє усвідомити задачу, а другий — успішно реалізувати її вирішення. Технічні характеристики Atlas`у взяті з джерела [22]:

- вага: 82 кг;
- зріст: 180 см;
- швидкість пересування: 1.5 м/с;
- кількість степенів свободи: 28 (по 6 для кожної ноги, по 6 для кожної руки, 3 для спини, 1 для шиї);
- гідравлічні приводи;
- невеликий гідравлічний силовий агрегат на 5 кг потужністю 5 кВт, котрий живиться від батареї, розрахованої на 1400 Вт*год;
- корисне навантаження: 11 кг;
- деякі деталі виконані на 3D принтері;
- сенсорна навігація реалізована за допомогою технологій LIDAR та стерео бачення;
- управління роботом зі спеціального дистанційного пульта керування, однак усі обчислення та регуляційні моменти він виконує самостійно.

Atlas призначений, передусім, для використання у небезпечних ситуаціях, де необхідно асистувати рятувальникам або виконувати рятувальні роботи на атомних електростанціях, у місцях ризику радіаційного зараження тощо [23].

Доказом корисності Atlas`у в рятувальних роботах є друге місце у конкурсі 2015-го року від DARPA, де робот успішно виконав наступні 8 завдань (інформація з джерела [22]):

1. Управління вантажною машиною.
2. Рух нерівною поверхнею зі щебеню.
3. Прибирання об`єкту, котрий блокує шлях.
4. Відкриття дверей та проникнення у приміщення.
5. Здіймання промисловою драбиною та рух промисловою доріжкою.
6. Використання інструменту для пробиття через бетонну панель.
7. Локалізація клапану та його перекриття біля протікаючої труби.

8. Підключення пожежного шлангу до труби та увімкнення клапану.

Таким чином, Atlas наразі є найбільш мобільним, функціональним та акробатичним роботом у світі. Його можливості детектування перешкод та динамічна розробка алгоритму їх уникнення за допомогою геометричної сегментації поверхні є дійсно вражаючими. Такі відомості будуть корисними під час роботи над вдосконаленням функціональних можливостей обраного ДКР відповідно до теми магістерської дисертації.

1.5 Робот-вогнеборець SAFFiR

Shipboard Autonomous Firefighting Robot — робот-гуманоїд, розроблений 2015-го року зусиллями трьох американських лабораторій спеціально для боротьби з вогнем у вузьких приміщеннях військово-морських суден (рисунок 1.6) [24].

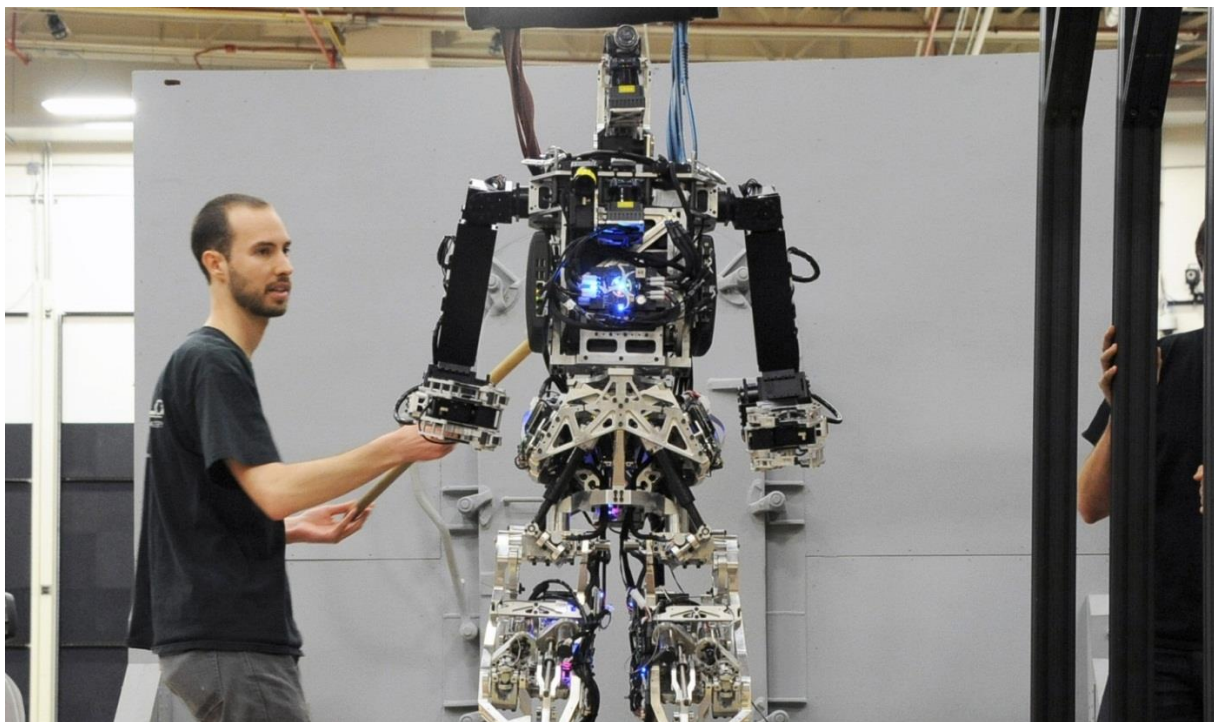


Рисунок 1.6 – Робот-вогнеборець SAFFiR [25]

Головною особливістю SAFFiR є його здатність працювати при високих температурах у замкненому просторі, що є критично важливим для задачі пожежогасіння у відділеннях морських кораблів. Це пов'язано з тим, що військово-морські кораблі мають у своїх приміщеннях різнокаліберну зброю та боеприпаси, а тому локалізувати та максимально швидко приборкати полум'я є фактично задачею життя та смерті, оскільки при досягненні вогнем кімнат з набоями може вибухнути весь корабель разом з екіпажем на борту.

Продумане програмне та апаратне забезпечення робота, високоробасна система стабілізації, пасивні послідовні пружини, а також ціла система датчиків дозволяє йому стійко витримувати природній крен судна при качках, залишаючись у вертикальному положенні, незважаючи на наявний кут нахилу. Інші сенсори дозволяють роботів орієнтуватися у просторі за умов сильного задимлення або занадто яскравого палахкотіння вогню та бачити навколишнє середовище [24].

Наразі робот вміє пересуватися штучними та природніми поверхнями, виявляти перешкоди, успішно оминати їх, а також самостійно розробляти маршрути, що будуть найбільш доцільними в конкретній ситуації, локалізувати джерело тепла, котре виділяє полум'я (знаходячись у сусідньому приміщенні за дверима), відкривати двері, розмотувати шланг, випускати струю води зі шлангу в його руках [24]. У «вільний час» робот зможе асистувати екіпажу морського судна, виконуючи перевірку кімнат на тепловтрати, або просто оглядати їх — це допоможе іншим морякам займатися більш важливими справами. Причому робот вміє все це пророблювати у неперервному режимі.

Технічні характеристики робота SAFFiR (взяті з джерела [26]):

- вага: 65 кг;
- зріст: 178 см;
- кількість степенів свободи: 33;
- час живлення від зовнішнього акумулятору: 30 хв;
- замість сервоприводів робот має штучні м'язи;
- зорова система: стандартне стерео, лазер, стерео-термальне зображення;
- наявність сильного захоплювача для тримання шлангу з водою;

- водонепроникний та термостійкий костюм задля уникання короткого замикання або критичних температур.

SAFFiR є гарним прикладом ДКР, котрий має високу стійкість, гарно налагоджену систему датчиків для визначення необхідних параметрів, важливих для локалізації проблеми, а також надійний алгоритм, котрий дозволяє оперативно обходити перешкоди та прокладати шлях до джерела полум'я.

1.6 Робот TORO

TORO — творіння дослідників з Інституту робототехніки та мехатроніки з Німеччини. Назва робота є акронімом від TOrque-controlled humanoid RObot, що натякає на спосіб управління роботом, котрий здійснюється на основі обертового моменту. Використовується для досліджень двоногої ходьби та її динаміки, а саме: робасна ходьба, взаємодія зі сходами, мультиконтактні операції [27]. Шлях становлення роботу TORO зображено на рисунку 1.7.



Рисунок 1.7 – Робот TORO та його еволюція [28]

TORO має наступні технічні характеристики [27]:

- вага: 76 кг;

- зріст: 174 см;
- кількість степенів свободи: 39 (по 6 для обох ніг, по 6 для обох рук, по 6 для обох долоней, 1 для торсу та 1 для шиї);
- номінальне корисне навантаження: 10 кг;
- енергопостачання: батарея на 1 год;
- швидкість пересування: 0.5 м/с або 1.8 км/год;
- датчики обертального моменту у майже всіх з'єднаннях;
- 2 датчики балансу (IMU-датчики);
- 2 датчики сили/обертального моменту у стопах;
- 2 стерео камери;
- 1 кольорова камера (RGB-D).

Головна особливість TORO — він управляється не через позиційне керування, а за рахунок обертального моменту. За ідеєю, даний підхід має забезпечити робастну взаємодію робота з навколишнім середовищем, підвищити його гнучкість в залежності від керуючого контролера.

У порівнянні з керуванням через задання конкретної позиції, керування через обертальний момент за рахунок цієї штучної гнучкості досягне більшого ступеню безпеки під час взаємодії з людиною та підвищить загальну стійкість робота у навколишньому середовищі [27].

TORO здатний ходити різними поверхнями, виявляти перешкоди та коректно з ними взаємодіяти, автоматично регулювати свій власний баланс (так званий еквілібріум), піднімати вантажі, втримувати рівновагу під час сторонніх втручань у роботу робота (наприклад, штовхання). Також, робот вміє балансувати лише на одній нозі.

Таким чином, дослідження та експерименти з роботом TORO потенційно здатні допомогти з майбутньою інтеграцією роботів у суспільство, де стійкість та коректна взаємодія з людьми буде у великій нагоді. Це також дуже цікавий екземпляр, який надає новий погляд на принципи керування роботом і при цьому чудово пояснює можливі переваги такого підходу.

1.7 Порівняльний аналіз розглянутих ДКР

Для більшої зручності зведемо усі важливі технічні характеристики до таблиці 1.1. Це надасть більшу наочність усієї зібраної інформації та дозволить одразу робити важливі для подальших досліджень висновки.

Таблиця 1.1 – Порівняння технічних характеристик розглянутих ДКР

Назва ДКР	ASIMO	DURUS	Atlas	SAFFiR	TORO
Вага, кг	48	100	82	65	76
Зріст, см	130	165	180	178	174
Час автономної роботи, год	0.5	Заявлено 8 год, експериментально - 4 год 52 хв	не вказано	0.5	1
Швидкість, м/с	0...2.5	0.22	1.5	≈ 0.15	0.5
Кількість степенів свободи	57	23	28	33	39
Корисне навантаження, кг	-	-	11	-	10
Особливості	Багато додаткової інтерактивності з людьми та об'єктами, уникнення перешкод	Найменш енергозатратний робот, висока ефективність ходьби	Паркур, унікальна техніка уникнення перешкод, багато-функціональність	*	**

* — здатність стійко витримувати природній крен судна при качках, залишаючись у вертикальному положенні незважаючи на наявний кут нахилу; орієнтація у просторі за умов сильного задимлення або занадто яскравого палахкотіння вогню; виявлення перешкод, успішне їх оминання, локалізація джерела тепла, котре виділяє полум'я (перебуваючи у сусідньому приміщенні), відкривання дверей, оперування шлангом з водою.

** — управління роботом не через позиційне керування, а через обертальний момент на приводах; TORO здатний ходити різними поверхнями, виявляти перешкоди та коректно з ними взаємодіяти, автоматично регулювати свій власний баланс, піднімати вантажі, втримувати рівновагу під час сторонніх втручань у роботу робота (наприклад, штовхання), балансувати на одній нозі.

З таблиці 1.1 можна побачити, що всі ДКР є різноманітними за усіма параметрами і дуже важко виділити когось одного та сказати, що він більш особливий, аніж інший робот.

Звичайно, найбільш помітними є ASIMO та Atlas — саме ці екземпляри показали максимальний спектр можливостей. Перший робот себе чудово проявив у взаємодії з людьми, а другий продемонстрував вражаючі акробатичні рухи, причому сама моторика цих рухів надзвичайно схожа на людську.

При цьому варто наголосити, що один важливий параметр не був включений до таблиці — це вартість. Він не був згаданий через те, що у багатьох розглянутих роботах вона не вказується. Однак, щодо декількох екземплярів є інформація:

1. Робот ASIMO у 2012-му коштував не менше ніж \$2 500 000 [29]. Наразі виробництво ASIMO призупинено — дослідники вирішили сконцентруватися на більш ефективних роботах, оскільки зіткнулися із сильною конкуренцією в обличчі американського Atlas.
2. Вартість робота Atlas невідома, але якщо врахувати його унікальну функціональність та вплив на припинення розробки серії ASIMO, робот має коштувати не менше за \$1 000 000.

Ціни на звичайних роботів-андроїдів для некомерційного використання розпочинаються з помітки \$20 000 [30]. Причому це ті роботи, які спрямовані швидше на відтворення емоцій, взаємодію з людиною через синтезовану мову і т. д.

Тобто, навіть за невелику ціну можна отримати лише обмеженого у функціональних можливостях робота, з чого випливає, що такі екземпляри не зможуть допомагати людям у небезпечних середовищах, навіть у перенесенні вантажу — а ціни достатньо високі. Це і є однією з головних проблем сучасних ДКР, яка заважає поширенню роботів. Ще одна — недостатній розвиток напрямку розробки роботів-гуманоїдів.

1.8 Спільна проблематика розглянутих ДКР

Так само, як і будь-яка інша область людської життєдіяльності, робототехнічний напрямок також має свої нюанси та актуальні проблеми. Особливо, якщо зважати на його відносну молодість, адже фундаментальні дослідження та перші реально вартісні результати даної галузі розпочалися трохи більше за півстоліття назад.

Уважно роздивляючись таблицю 1.1, а також попередньо наведені відомості про ДКР, можна помітити певні проблемні точки у всіх екземплярах, які не вдалося вирішити жодній команді розробників, або якщо і вдалося, то ціною обмеження інших функціональних можливостей робота. Саме тому розгляд таких проблем буде доцільним.

Першим йде енергоспоживання. Цей пункт є, напевно, максимально важливим для будь-якого робота, оскільки навіть найбільш мультифункціональні екземпляри будуть даремними, коли подача живлення раптово припиниться і робот буде змушений вимкнутися. Якщо говорити саме про небезпечні середовища, де робот має асистувати людині (зони радіоактивного, хімічного та біологічного зараження, пожежі і т. д.), виходить, що ризик зривання усієї операції є надзвичайно високим та матиме фатальні наслідки.

Розглянуті роботи показують, що максимальний час живлення – 1 год, а середній – 30 хв, що є досить поганим показником, якщо згадувати різні рятувні операції, котрі виконуються багато годин, а подекуди навіть і днів. Лише DURUS зміг протриматися близько 5 год завдяки двом батареям, які по чергово виснажувалися. Однак, цей робот має обмежену функціональність. DURUS надзвичайно енергоефективний у порівнянні з іншими, максимально наближено відтворює людську ходьбу, але це все, що він уміє. Такий робот матиме нульову корисність для команди рятувальників, ліквідаторів абощо.

Таке становище пов'язане з двома факторами: 1. Недостатньо ефективні енергоносії. 2. Високе споживання енергії зі сторони робототехнічної системи.

Можна замінити автономне живлення на постійне за допомогою підключених кабелів до генератору, однак це не тільки не вирішить проблему високого енергоспоживання, а також й створить додаткові — кабелі будуть суттєво заважати рухові робота та будуть загалом досить проблематичними. Такий варіант підійде лише для тих екземплярів, котрі знаходяться на стадії лабораторних випробувань, але ніяк не для реальної життєвої динаміки.

Другий пункт — виробництво. Досить велика кількість розробників у роботобудівництві використовує двигунки, двигуни та приводи, які роблять робота жорстким, ненадійним та нестійким. Це нераціональний підхід, з яким необхідно боротися створенням більш досконалих робототехнічних комплектуючих та роблячи їх легкодоступнішими. На щастя, даний напрямок вже існує і називається він «soft robotics» [31]. Спрямований на розробку більш гнучких та зручних компонентів для діяльності у сфері роботобудування.

Надійний штучний інтелект (ШІ) — третій важливий пункт. Не дивлячись на широке розповсюдження технік машинного навчання та ряди успішних впроваджень ШІ, ці технології досі лишаються складними. Навіть робот Atlas для того, щоб навчитися здійснювати певну акробатичну взаємодію із зовнішнім середовищем, пройшов через безліч ітерацій із завантаженням тонни даних, аби вміти вдало виконувати покладену на нього роботу у заданих дослідниками умовах.

Навколишнє середовище є нестабільним, а тому навіть вдалі результати в лабораторних умовах не дають гарантій, що робот у незнайомій ситуації зможе коректно імпровізувати. Саме тому нинішні технології ШІ не є надійними та мають обмежений діапазон реального застосування.

Четвертий пункт — навігація у навколишньому середовищі. Її проблема полягає в її обмеженості буквально декількома метрами, чого недостатньо для всеохоплюючого автоматичного створення картографії місцевості, у якій належить функціонувати роботу. Той же Atlas має «зону видимості» близько 1.5 м, а японський ASIMO користується вже вбудованою картою місцевості.

Розглянуті проблеми є частиною всезагальних проблем робототехніки, котрі можна спостерігати на рисунку 1.8.

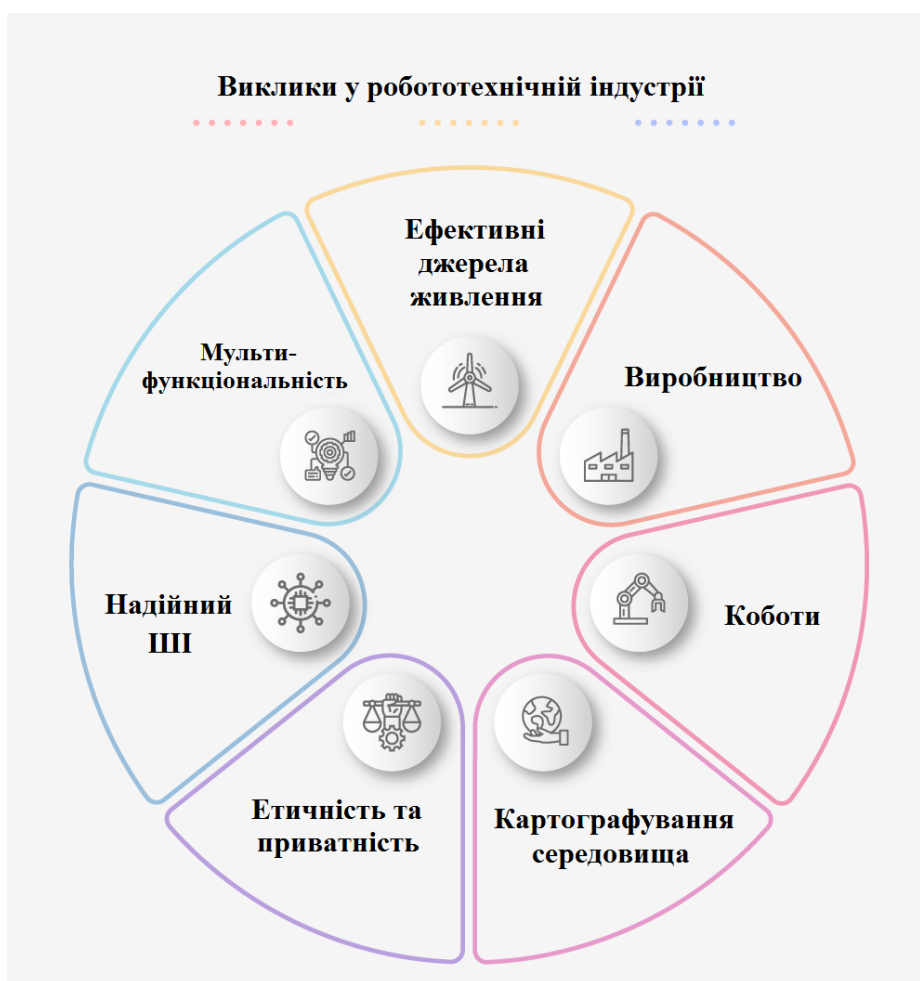


Рисунок 1.8 – Спектр проблем у робототехнічній індустрії [32]

Висновки до розділу 1

Розглянута п'ятірка роботів-гуманоїдів підтвердила актуальність даної тематики у робототехнічній галузі.

При цьому було визначено декілька проблемних точок, котрі стримують плідний розвиток робототехніки. В основному ці точки лежать у площині надмірності: надмірне споживання електроенергії, великі розумові та часові витрати команди розробників, використання нераціональних компонентів, котрі створюють зайвий шум, нестабільність та «грубість» ходьби, дають надмірність вазі робота; недосконалі техніки реалізації ІІІ та машинного навчання тощо. Також, велика вартість роботів свідчить як про великі матеріальні і нематеріальні зусилля, покладені на створення хоча б одного екземпляру ефективного робота, так і про неможливість виходу ДКР на масові ринки. Деякі з цих позицій були помічені також і у розглянутих екземплярах.

Подібне становище сигналізує про необхідність вдосконалення технік розроблення ДКР (як програмне, так і апаратне забезпечення), що паралельно вестиме до зниження сумарної вартості робота і сприятиме його скорішому виходу на масовий ринок.

З огляду на необхідність спрощення технік навчання робота, які пов'язані з подоланням перешкод, наступні кроки магістерської дисертації доцільно направити саме в сторону спрощення алгоритмів взаємодії ДКР з оточуючими об'єктами.

2 РОЗРОБЛЕННЯ ВДОСКОНАЛЕНОЇ МОДЕЛІ РОБОТА У СЕРЕДОВИЩІ MATLAB/SIMULINK

2.1 Опис математичної моделі антропоморфного механізму

Для того, щоб зрозуміти принципи переміщення ДКР, доцільно розглянути динаміку переміщення двоногих антропоморфних істот загалом. Вона описана у [33], а також гарно подана у більш стиснутому та лаконічному вигляді у [34]. Наведемо динаміку і у даній дисертації, користуючись джерелом [34].

На рисунку 2.1 у двовимірній системі координат зображено двоногий механізм, котрий здатний відтворювати людську локомоцію.

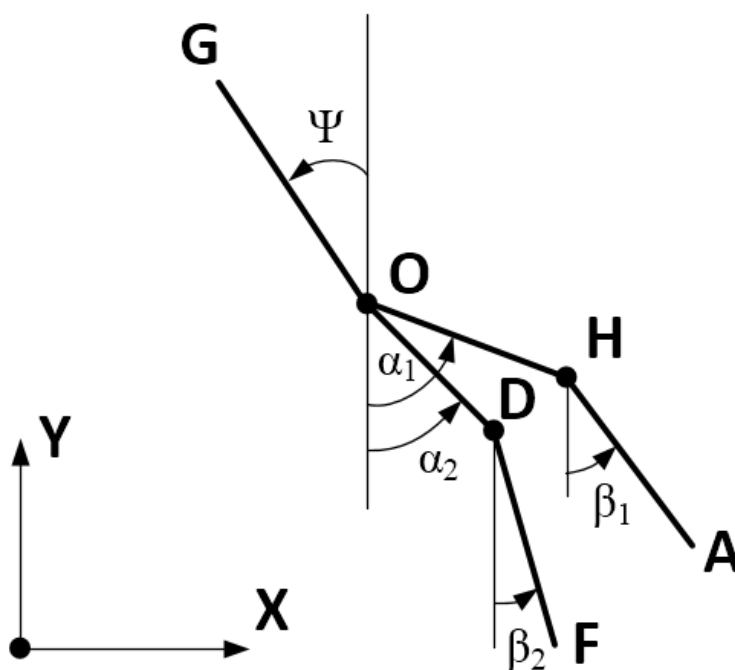


Рисунок 2.1 – Плоский механізм, здатний відтворювати людську локомоцію

З рисунку 2.1 маємо: п'ять шарнірних ланок, котрі з'єднані та мають певну вагу, корпус OG; OD та OH – ланки стегна, DF та HA – ланки гомілки. Шарнір O з'єднує корпус з двома кінцівками OF та OA і називається тазостегновий суглоб,

шарніри D та H з'єднують голені та гомілки і називаються колінними суглобами. Кожен шарнір розглядаємо як ідеальний з повною відсутністю тертя.

XY – нерухома двовимірна система координат, де вісь Y відкладена вертикально, а X має горизонтальне положення відповідно до рисунку 2.1. Механізм має 7 степенів свободи. Перейдемо до позначення узагальнених координат: x , y – координати тазостегнового суглобу O, ψ – кут між корпусом та вертикаллю, α_1 та α_2 – кути, між стегнами та вертикаллю, β_1 та β_2 – кути між вертикаллю та двома гомілками.

Для складання рівняння динаміки п'ятиланкового механізму користуватимемося рівняннями Лагранжа другого порядку (2.1):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}_s} \right) - \frac{\partial L}{\partial z_s} = Q_s, \quad s = \overline{1, n}, \quad (2.1)$$

де z_s – узагальнена координата;

Q_s – узагальнена неконсервативна сила;

$L = E_K - E_{\Pi}$ – функція Лагранжа, котра дорівнює різниці кінетичної енергії (E_K) та потенціальної (E_{Π});

n – кількість степенів свободи механізму, що дорівнює 7 у даному випадку.

Для знаходження кінетичної енергії кожної ланки необхідно використовувати формулу (2.2):

$$E_K = \frac{1}{2} [mv^2 + 2m((v \times \omega), \rho) + \theta \omega^2], \quad (2.2)$$

де v – швидкість полюсу (або ж фіксованої точки ланки);

ω – кутова швидкість обертання ланки;

ρ – радіус-вектор центру мас ланки, що починається у полюсі;

m – маса ланки;

θ – момент інерції ланки відносно фіксованої точки ланки.

Точка О буде взята в якості полюсу для ланок OG, OH, OD. Корпус механізму позначимо як $E_{K(OG)}$. Тоді його кінетичну енергію можна знайти з виразу (2.3):

$$E_{K(OG)} = \frac{1}{2} [m_k(\dot{x}^2 + \dot{y}^2) - 2K_r\psi(\dot{x}\cos\psi + \dot{y}\sin\psi) + J\dot{\psi}^2], \quad (2.3)$$

де K_r – добуток m_k та r ;

m_k – маса корпусу;

r – відстань від точки О до центр мас корпусу;

J – момент інерції корпусу відносно тазостегнового суглобу О.

В такому випадку визначення кінетичної енергії стегна ОН буде здійснюватися за формулою (2.4):

$$E_{K(OH)} = \frac{1}{2} [m_a(\dot{x}^2 + \dot{y}^2) + 2m_a a \dot{\alpha}_1(\dot{x} \cos \alpha_1 + \dot{y} \sin \alpha_1) + J_a^0 \dot{\alpha}_1^2], \quad (2.4)$$

де m_a – маса стегна;

a – відстань від тазостегнового суглобу до центру мас стегна;

J_a^0 – момент інерції стегна відносно тазостегнового суглобу О.

Кінетичну енергію стегна OD ($E_{K(OD)}$) можна визначити через заміну кута α_1 на α_2 у виразі (2.4). Виконуючи обчислення кінетичної енергії гомілки НА, беремо у якості полюсу точку Н, котра відіграє роль коліна. Тоді кінетичну енергію гомілки НА можна буде знайти з виразу (2.5):

$$E_{K(HA)} = \frac{1}{2} \{ m_b[\dot{x}^2 + \dot{y}^2 + 2\dot{\alpha}_1 L_a(\dot{x} \cos \alpha_1 + \dot{y} \sin \alpha_1) + \dot{\alpha}_1^2 L_a^2] + 2K_b \dot{\beta}_1[\dot{x} \cos \beta_1 + \dot{y} \sin \beta_1 + \dot{\alpha}_1 L_a \cos(\alpha_1 - \beta_1)] + J_b \dot{\beta}_1^2 \}, \quad (2.5)$$

де K_b є добутком m_b та b ;

m_b – маса гомілки;

b – відстань від точки Н до центру мас гомілки;

L_a – довжина стегна;

J_b – момент інерції гомілки відносно колінного суглобу Н.

Кінетичну енергію гомілки DF ($E_{K(DF)}$) можна отримати, замінивши кути α_1, β_1 на кути α_2, β_2 . Наразі з використанням співвідношень (2.3) – (2.5) можливо записати формулу для кінетичної енергії всієї п'ятиланкової конструкції (2.6):

$$\begin{aligned} E_K = & E_{K(OG)} + E_{K(OH)} + E_{K(OD)} + E_{K(HA)} + E_{K(DF)} = \\ = & \frac{1}{2}M(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}J\dot{\psi}^2 - K_r\psi(\dot{x}\cos\psi + \dot{y}\sin\psi) + \\ + & \sum_{i=1}^2 \left[\frac{1}{2}J_a\dot{\alpha}_i^2 + \frac{1}{2}J_b\dot{\beta}_i^2 + K_a\dot{\alpha}_i(\dot{x}\cos\alpha_i + \dot{y}\sin\alpha_i) + \right. \\ & \left. + K_b\dot{\beta}_i(\dot{x}\cos\beta_i + \dot{y}\sin\beta_i) + J_{ab}\dot{\alpha}_i\dot{\beta}_i\cos(\alpha_i - \beta_i) \right], \end{aligned} \quad (2.6)$$

де $M = m_k + 2m_a + 2m_b$ – маса п'ятиланкового механізму;

K_a – сума добутку $m_a a$ та добутку $m_b L_a$;

J_a – сума моменту інерції стегна відносно тазостегнового суглобу О (J_a^0) та добутку $m_b L_a^2$;

J_{ab} – добуток $K_b L_a$, котрий є рівним добутку $m_b b L_a$.

Щоб обраховувати потенціальну енергію E_{Π} п'ятиланкового механізму, слід використати формулу (2.7):

$$E_{\Pi} = g \left\{ m_k(y + r \cos\psi) + \sum_{i=1}^2 [m_a(y - a \cos\alpha_i) + m_b(y - L_a \cos\alpha_i - b \cos\beta_i)] \right\},$$

або, якщо слідувати встановленим нормам

$$E_{\Pi} = g \left[M_y + K_r \cos\psi - \sum_{i=1}^2 (K_a \cos\alpha_i + K_b \cos\beta_i) \right], \quad (2.7)$$

де g – прискорення вільного падіння (9.8 м/с^2).

Необхідно визначити внутрішні неконсервативні сили, котрі впливають на задану п'ятиланкову конструкцію. Оновлене зображення механізму можна спостерігати на рис. 2.2. Позначимо як μ_1 та μ_2 моменти сил, які присутні в колінних суглобах. Тоді q_1 та q_2 – моменти сил, котрі прикладені між корпусом механізму та його стегнами.

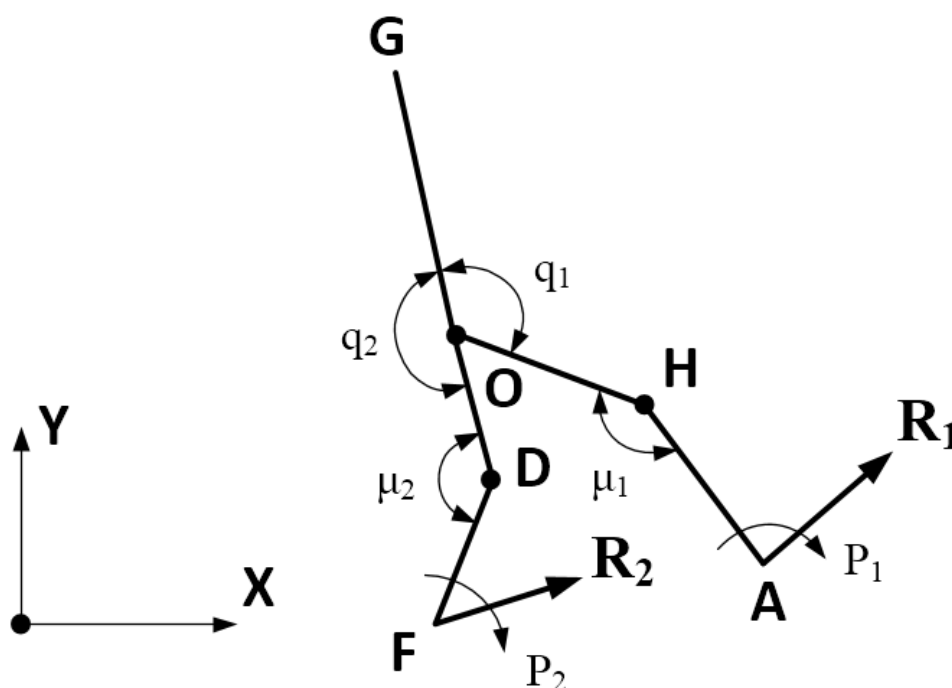


Рисунок 2.2 – Зображення сил, котрі впливають на п'ятиланковий механізм

Зовнішні сили реакцій поверхні, на яку опирається конструкція, позначені R_1 та R_2 та прикладені до кінців ніг (точки A , F); P_1 та P_2 являються моментами зовнішніх сил, котрі є прикладеними до гомілок. Сили R_1 та R_2 представлені горизонтальними (R_{1x} , R_{2x}) та вертикальними (R_{1y} , R_{2y}) складовими. Вважається, що усі шарніри, які з'єднують ланки, є ідеальними, а моменти μ_1 , μ_2 , q_1 , q_2 , P_1 , P_2 в шарнірах створюються за рахунок приводів.

Розпочнемо складати рівняння руху. Для цього необхідно знати 7 узагальнених сил Q_x , Q_y , Q_ψ , $Q_{\alpha 1}$, $Q_{\alpha 2}$, $Q_{\beta 1}$, $Q_{\beta 2}$, які можна знайти з виразу елементарної роботи δW усіх сил, що були прикладені до заданої системи (3.8):

$$\begin{aligned}
\delta W = & (R_{1x} + R_{2x})\delta x + (R_{1y} + R_{2y})\delta y - (q_1 + q_2)\delta\psi + \\
& + \sum_{i=1}^2 [(q_i - \mu_i)\delta\alpha_i + (\mu_i - P_i)\delta\beta_i + R_{ix}\delta(L_a\sin\alpha_i + L_b\sin\beta_i) - \\
& - R_{iy}\delta(L_a\cos\alpha_i + L_b\cos\beta_i)] = \sum_{i=1}^2 [R_{ix}\delta x + R_{iy}\delta y - q_i\delta\psi + \\
& + (q_i - \mu_i + R_{ix}L_a\cos\alpha_i + R_{iy}L_a\sin\alpha_i)\delta\alpha_i + \\
& + (\mu_i - P_i + R_{ix}L_b\cos\beta_i + R_{iy}L_b\sin\beta_i)\delta\beta_i], \tag{2.8}
\end{aligned}$$

де L_b є довжиною гомілки. З розписаного виразу (2.8) знаходимо шукану сімку узагальнених сил (2.9.1)-(2.9.5):

$$Q_x = R_{1x} + R_{2x}, \tag{2.9.1}$$

$$Q_y = R_{1y} + R_{2y}, \tag{2.9.2}$$

$$Q_\psi = -(q_1 + q_2), \tag{2.9.3}$$

$$Q_{\alpha_i} = -\mu_i + q_i + L_a(R_{ix}\cos\alpha_i + R_{iy}\sin\alpha_i), \quad i = 1, 2, \tag{2.9.4}$$

$$Q_{\beta_i} = \mu_i - P_i + L_b(R_{ix}\cos\beta_i + R_{iy}\sin\beta_i), \quad i = 1, 2. \tag{2.9.5}$$

Користуючись формулами (2.6), (2.7), (2.9.1)-(2.9.5), можна представити $\partial L/\partial \dot{z}_s$, $\partial L/\partial z_s$, і вираз Лагранжа другого порядку (2.1) загалом у наступному вигляді (2.10.1)-(2.10.5) (детальніше у джерелі [33]):

$$M\ddot{x} - K_r\ddot{\psi}\cos\psi + K_a\ddot{\alpha}_1\cos\alpha_1 + K_a\ddot{\alpha}_2\cos\alpha_2 + K_b\ddot{\beta}_1\cos\beta_1 +$$

$$+ K_b \ddot{\beta}_2 \cos \beta_2 + K_r \dot{\psi}^2 \sin \psi - K_a \dot{\alpha}_1^2 \sin \alpha_1 - K_a \dot{\alpha}_2^2 \sin \alpha_2 - \\ - K_b \dot{\beta}_1^2 \sin \beta_1 - K_b \dot{\beta}_2^2 \sin \beta_2 = R_{1x} + R_{2x}, \quad (2.10.1)$$

$$M \ddot{y} - K_r \ddot{\psi} \sin \psi + K_a \ddot{\alpha}_1 \sin \alpha_1 + K_a \ddot{\alpha}_2 \sin \alpha_2 + K_b \ddot{\beta}_1 \sin \beta_1 + \\ + K_b \ddot{\beta}_2 \sin \beta_2 - K_r \dot{\psi}^2 \cos \psi + K_a \dot{\alpha}_1^2 \cos \alpha_1 + K_a \dot{\alpha}_2^2 \cos \alpha_2 + \\ + K_b \dot{\beta}_1^2 \cos \beta_1 + K_b \dot{\beta}_2^2 \cos \beta_2 = R_{1y} + R_{2y} - Mg, \quad (2.10.2)$$

$$-K_r \ddot{x} \cos \psi - K_r \ddot{y} \sin \psi + J \ddot{\psi} - g K_r \sin \psi = -q_1 - q_2, \quad (2.10.3)$$

$$K_a \ddot{x} \cos \alpha_i + K_a \ddot{y} \sin \alpha_i + J_a \ddot{\alpha}_i + J_{ab} \ddot{\beta}_i \cos(\alpha_i - \beta_i) + \\ + g K_a \sin \alpha_i + J_{ab} \dot{\beta}_i^2 \sin(\alpha_i - \beta_i) = \\ = -\mu_i + q_i + L_a R_{ix} \cos \alpha_i + L_a R_{iy} \sin \alpha_i, \quad i = 1, 2, \quad (2.10.4)$$

$$K_b \ddot{x} \cos \beta_i + K_b \ddot{y} \sin \beta_i + J_{ab} \ddot{\alpha}_i \cos(\alpha_i - \beta_i) + J_b \ddot{\beta}_i + \\ + g K_b \sin \beta_i - J_{ab} \dot{\alpha}_i^2 \sin(\alpha_i - \beta_i) = \\ = \mu_i - P_i + L_b R_{ix} \cos \beta_i + L_b R_{iy} \sin \beta_i, \quad i = 1, 2. \quad (2.10.5)$$

Визначення (2.10.1)-(2.10.5), які характеризують рух механізму, можна подати в іншому вигляді зі врахуванням позначень, наведених у книзі Формальського [33], у зручній векторно-матричній формі (2.11):

$$B(z) \ddot{z} + g A f_1(z) + D(z) f_2(\dot{z}) = C(z) w, \quad (2.11)$$

де z , $f_1(z)$ та $f_2(\dot{z})$, w мають наступний вигляд (2.12):

$$z = \begin{pmatrix} x \\ y \\ \psi \\ \alpha_1 \\ \alpha_2 \\ \beta_1 \\ \beta_2 \end{pmatrix}, \quad f_1(z) = \begin{pmatrix} 0 \\ 1 \\ \sin\psi \\ \sin\alpha_1 \\ \sin\alpha_2 \\ \sin\beta_1 \\ \sin\beta_2 \end{pmatrix}, \quad f_2(\dot{z}) = \begin{pmatrix} 0 \\ 0 \\ \dot{\psi}^2 \\ \dot{\alpha}_1^2 \\ \dot{\alpha}_2^2 \\ \dot{\beta}_1^2 \\ \dot{\beta}_2^2 \end{pmatrix},$$

$$w^* = (\mu_1, \mu_2, q_1, q_2, P_1, P_2, R_{1x}, R_{1y}, R_{2x}, R_{2y}), \quad (2.12)$$

де зірочка вказує на транспонування. З виглядом матриць A , $B(z)$, $D(z)$ розміром 7 на 7, а також матрицею $C(z)$ розміром 7 на 10 можна познайомитися у вже згаданій праці Формальського [33].

Важливо мати на увазі, що $B(z)$ більша за нуль – це додатньо визначена матриця кінетичної енергії. З цього випливає (2.13):

$$E_K = E_K(z, \dot{z}) = \frac{1}{2} \dot{z}^* B(z) \dot{z}. \quad (2.13)$$

Таким чином, вираз (2.11) можна прийняти відносно вектору старших похідних (2.14):

$$\ddot{z} = -gB^{-1}(z) A f_1(z) - B^{-1}(z) D(z) f_2(\dot{z}) + B^{-1}(z) C(z) w. \quad (2.14)$$

В даному виразі A виступає діагональною постійною матрицею, а елементи матриці $D(z)$ являють собою суть символів Кристоффеля першого роду для матриці $B(z)$.

Отже, через рівняння (2.11) можна описувати динаміку як одноопорного руху (ходьба людини), так і двоопорного (наприклад, під час використання лиж, при ходьбі слизькою місцевістю взимку тощо) — треба тільки у відповідності включати до динаміки R_1 або R_2 , чи і R_1 , і R_2 .

2.2 Розроблення структурної схеми авторської моделі двоногого робота, виконаної у MATLAB/Simulink

Розглянута у пункті 2.1 динаміка ходьби дозволяє зрозуміти основні засади функціонування антропоморфних механізмів з точки зору математичних і фізичних описових засобів. Віртуальне середовище MATLAB/Simulink в свою чергу дозволяє приховати складну математику і працювати зі зрозумілими блоками, за допомогою яких можна легко будувати та досліджувати не тільки ДКР, а й більш складні механізми, як-от машини, двигуни літаків тощо.

MATLAB – це пакет програм, котрі дозволяють вирішувати різноманітні інженерні задачі. Одна із його особливостей – однойменна високорівнева інтерпретована мова програмування, яка своєю лаконічністю, простотою та семантикою нагадує мову Python. Однак, візитною карткою MATLAB`у є саме його середовище інтерактивного моделювання – Simulink. Саме цей інструмент допоможе у зручному вигляді моделювати ДКР та проводити моніторинг важливих характеристик.

Для роботи над вдосконаленням функціональних можливостей моделі ДКР була взята вже готова модель, розроблена Себастьяном Кастро (Sebastian Castro, далі – «автор») – одним із популяризаторів MATLAB-продуктів. Він записав серію відеороликів і виклав їх на офіційному сайті MATLAB [35], в яких пояснює загальну базу функціонування крокуючих роботів. Додатково автор розмістив на сервісі для хостингу ІТ-проектів GitHub усі файли, завантаження яких дозволяє отримати авторську модель ДКР, що має коректно працювати у MATLAB версії R2017a [36].

На рисунку 2.3 наведена структурна схема з авторським ДКР, котра містить усі важливі структурні компоненти створеної автором моделі робота.

Із цієї структурної схеми можна легко визначити призначення та спосіб взаємодії усіх складових моделі ДКР.

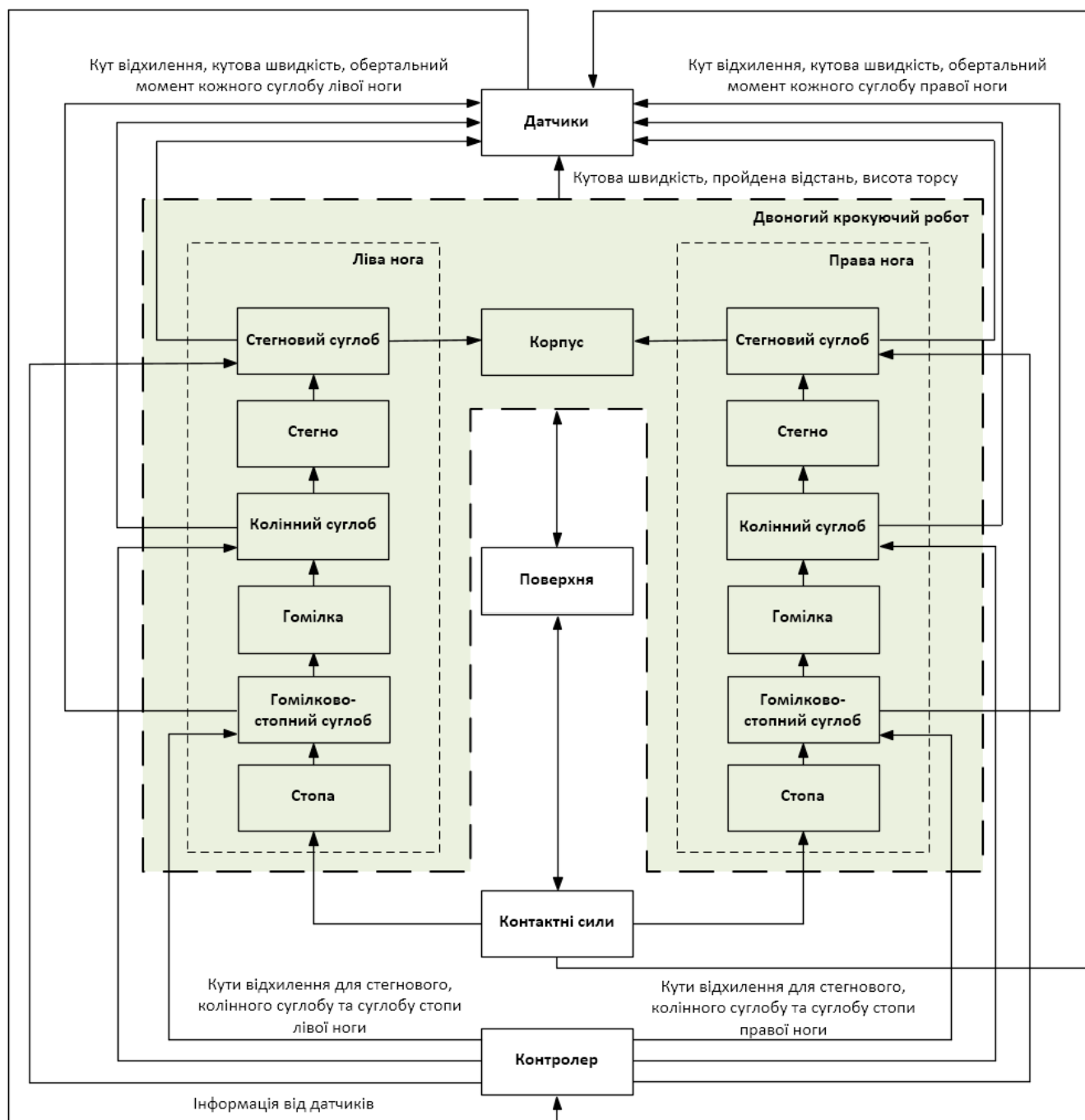


Рисунок 2.3 – Структурна схема моделі авторського ДКР, розробленої у середовищі
MATLAB R2017a

Відповідно до рисунку 2.3: завдяки стегновим суглобам до корпусу робота кріпиться права та ліва нога подібно до усіх антропоморфних організмів. Кожна нога утворена послідовним з'єднанням стопи з гомілковостопним суглобом,

гомілковостопного суглоба з гомілкою, гомілки з колінним суглобом, колінного суглоба зі стегном, і, нарешті, стегна зі стегновим суглобом.

До стоп приєднані спеціальні блоки, які відповідають за контактні сили між стопами та поверхнею, якою пересувається робот — дані блоки забезпечують їх логічну взаємодію. Ці сили зчитуються з тих самих блоків середовища Simulink, котрі задані сили фактично і генерують. Проте в даній ситуації для більшої зрозумілості структурної схеми буде введено поняття «датчики», в яке будуть інкапсульовані усі нюанси, пов'язані зі зчитуванням тих чи інших даних.

Також, зчитуванню піддаються параметри гомілковостопного, колінного та стегнового суглобів, а саме: кут відхилення стопи, гомілки та стегна (відповідно), їх кутові швидкості та обертальні моменти. При цьому сигнали руху подаються через своєрідний контролер, котрий у середовищі Simulink представлено блоками, що видають кути необхідних відхилень кожного суглоба робота через кожний заданий інтервал часу.

Крім того, зчитуються дані і з робота в цілому — його кутова швидкість, пройдена ним відстань за час моделювання і висота торсу над поверхнею при ходьбі. Ці дані просто фіксуються контролером і на поведінку робота не впливають. Лише його падіння спричинить зупинку моделювання.

Також, сам ДКР пов'язаний з поверхнею певними налаштуваннями, як-от висота робота над поверхнею, відстань від початку координат тощо.

2.3 Структурна схема модернізованої моделі двоногого крокуючого робота, виконаної у MATLAB/Simulink

Передусім важливо сказати, що модернізована модель двоногого крокуючого робота отримала назву Darth EvadeR. Сенс даної назви простий: evader з англ. перекладається як «уникач», що натякає на здібності робота до уникнення перешкод.

Структурна схема двоногого крокуючого робота Darth EvadeR представлена на рисунку 2.4.

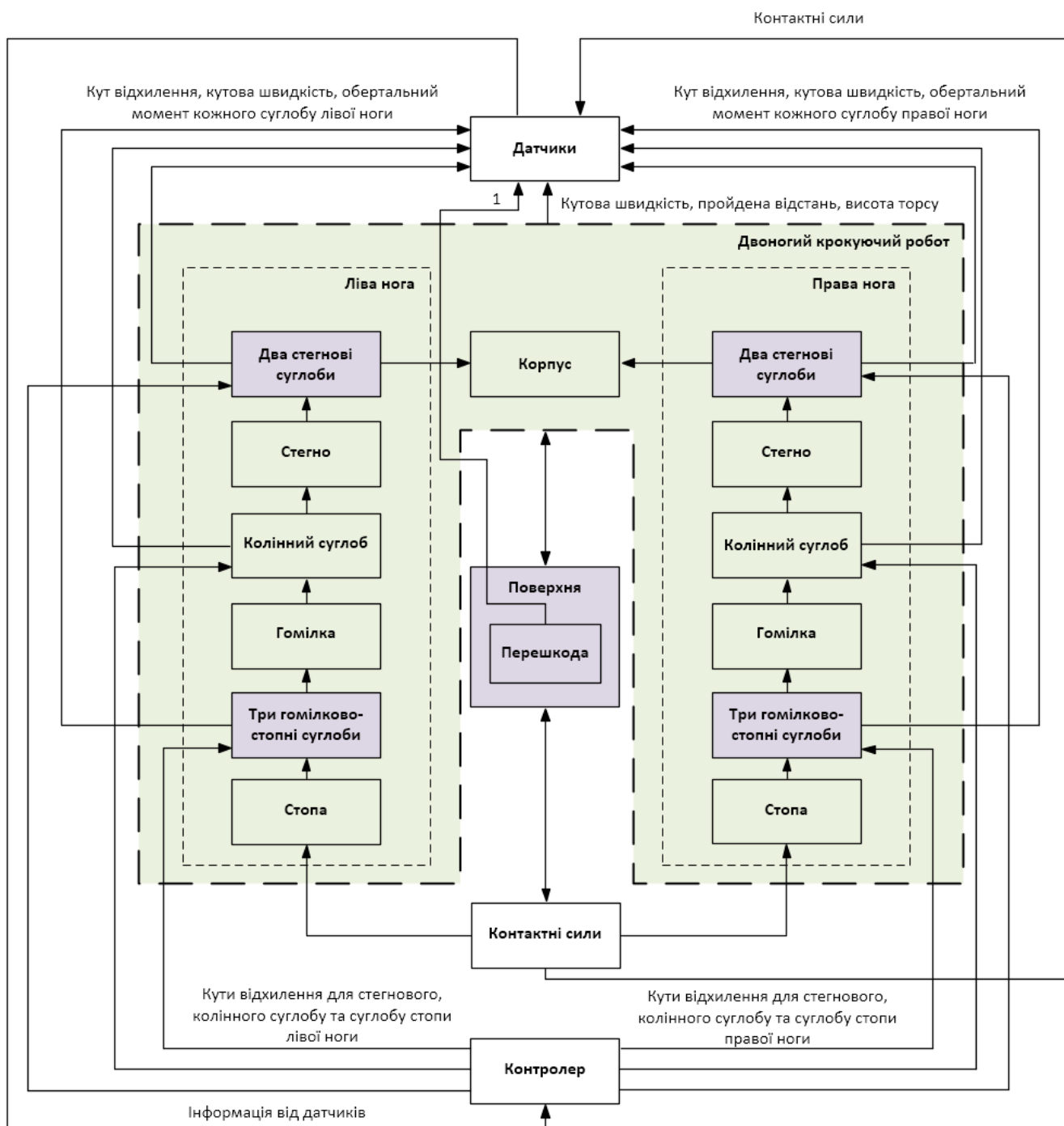


Рисунок 2.4 – Структурна схема моделі робота Darth EvadeR

У порівнянні з моделлю оригінального авторського робота, Darth EvadeR має наступні структурні відмінності:

1. Кількість степенів свободи. Авторський робот мав 6 степенів свободи, котрі відповідали загальній кількості суглобів: 2 гомілково-стопних, 2

колінних, 2 стегнових (усі функціонують вздовж осі X, або як це прийнято називати по-іншому – тангаж). Darth EvadeR має 12 степенів свободи: до авторських додалися по 2 гомілково-стопних суглоби для того, щоб робот міг розміщувати свою стопу у просторі не тільки за віссю X, але й за осями Y та Z (рискання та крен, відповідно); також, було додано по 1-му стегновому суглобу, аби робот міг здійснювати рух стегном вбік (вісь Y, рискання). Детальніше щодо положення Darth EvadeR`а відносно даних осей буде розписано нижче.

2. Додавання перешкоди на шляху робота, а також реалізація її фіксування датчиком робота. Цифрою 1 на рисунку 3.2 позначено інформацію, котра сприймається датчиком щодо відстані робота до перешкоди.

На рисунку 2.5 зображена модель Darth EvadeR`а разом із позначеними осями.

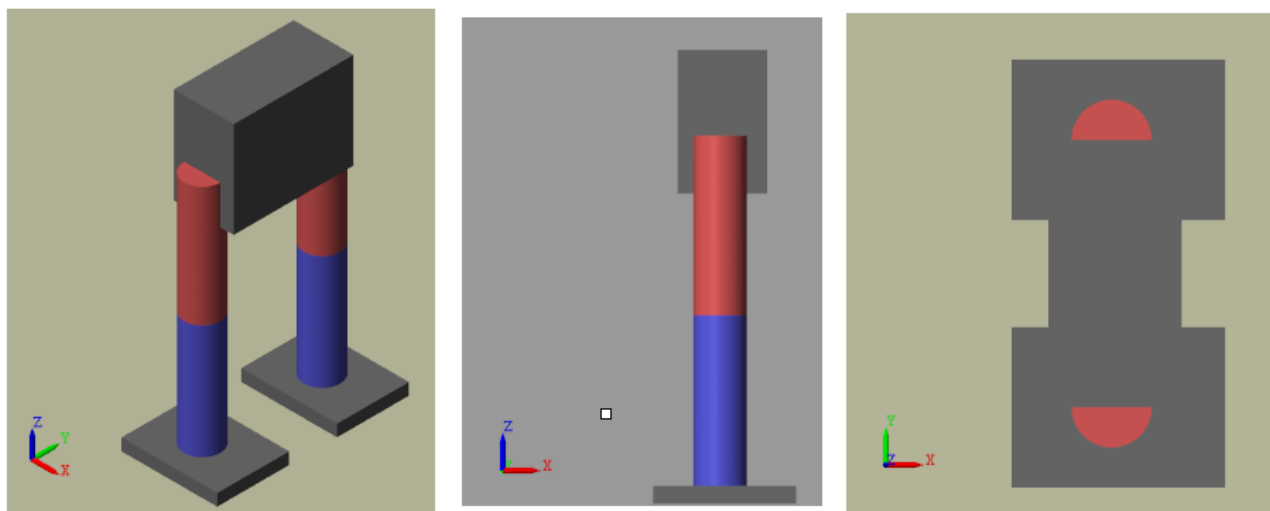


Рисунок 2.5 – Модель робота Darth EvadeR з позначеними ліворуч осями:
зображення ліворуч – ізометричний вигляд, центр – вигляд збоку, праворуч – вигляд згори

Співставляючи розглянуті структурні схеми (рис. 2.3 та 2.4), пояснення до них, а також рисунок 2.5, легко бачити, що суглоби, орієнтовані на роботу вздовж осі X, забезпечують рух робота вздовж поверхні; суглоби, пов'язані з віссю Y

забезпечують бокові рухи робота; суглоби, пов'язані з віссю Z відповідають за рухи стопи виду «на себе – від себе».

2.4 Будова вдосконаленої моделі робота Darth EvadeR у середовищі MATLAB/Simulink

Як вже раніше зазначалося, модель виконана у середовищі MATLAB R2017a. Наведені у розділі 3 структурні схеми дозволяють скласти загальну картину моделі двоногого крокуючого робота. Доцільним буде наведення та пояснення блоків, з яких складається сам робот.

Для побудови робота використовувалася бібліотека Simscape Multibody. На рисунку 2.6 наведені основні блоки, на яких побудовано відображення та функціонування правої ноги.

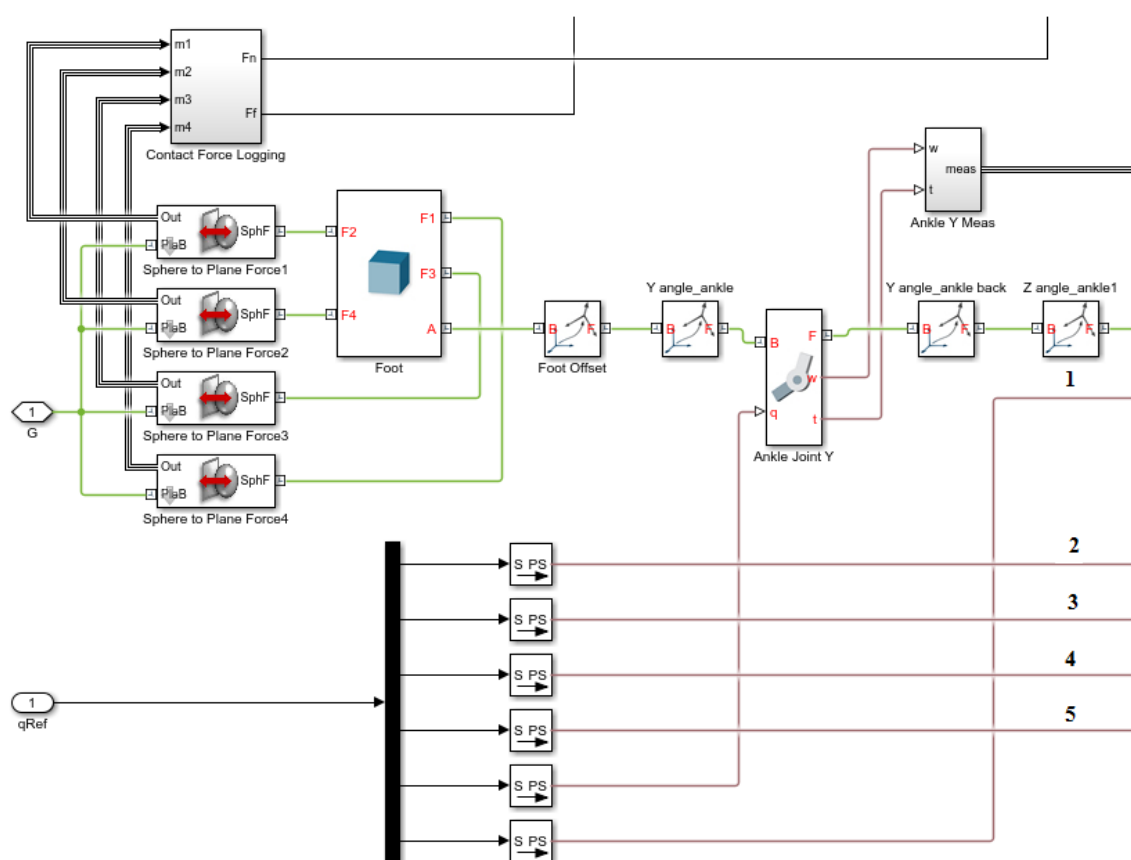


Рисунок 2.6 – Перший фрагмент моделі правої ноги робота Darth EvadeR

На рисунку 2.6 блок Foot відповідає за фізичне відображення стопи, тим самим визначаючи її присутність у графічному вікні Mechanics Explorer, котре з'являється під час запуску моделювання. Також, Foot під'єднано до чотирьох блоків Sphere to Plane Force – вони визначають контактну взаємодію стоп з поверхнею, якою рухається робот. Кожен блок фактично розміщує в кожному з чотирьох кутів стопи невелику сферу, яка і відповідає за контактну реалізацію. Фактично, Sphere to Plane Force визначає спосіб взаємодії чотирьох сфер на кутах стопи з плоскою площиною. При цьому необхідно вказати такі параметри як: жорсткість контакту (Н/м), їх демпфування (Н/(м/с)), закон сили, закон тертя, коефіцієнт кінетичного тертя, коефіцієнт статичного тертя тощо. Значеннях цих параметрів були підібрані експериментальним шляхом зі врахуванням авторських значень та коментарів.

Contact Force Logging – підсистема, яка збирає інформацію про значення нормальної сили та сили тертя.

Через блоки Foot Offset та Y angle_ankle блок Foot кріпиться до суглобу Ankle Joint Y. Foot Offset та Y angle_ankle – це не що інше, як перейменований блок Rigid Transform; його функція – визначення фіксованої 3D трансформації між двома кадрами (у випадку Foot Offset – між Foot та Y angle_ankle, у випадку Y_angle_ankle – між Foot Offset та Ankle Joint Y). Тобто, блоки Rigid Transform відображають орієнтацію осі наступного кадру (блоку). Порти B та F представляють базовий та наступний кадри відповідно. Це необхідно аби, наприклад, стопа могла коректно рухатися відносно гомілки за участі гомілково-стопного суглобу.

Таким чином, блок Y angle_ankle регламентує обертання стопи віссю Y для гомілковостопного суглобу Ankle Joint Y на рисунку 2.6. Y_angle_ankle back «закриває» зону дії руху суглобу віссю Y, котру «відкрив» Y angle_ankle.

Блок Ankle Y Meas знімає покази обертового моменту та кутової швидкості блоку-суглобу Ankle Joint Y. Надалі усі схожі блоки на інших фрагментах схеми будуть функціонувати за тими ж принципами, що і вже описані блоки з рисунку 2.6.

qRef – це вхідний порт, котрий постачає усі шість суглобів моделі правої ноги необхідними значеннями відхилення у радіанах кожену 0.1333 с. Цикл ходьби робота

триває 0.8 с і кожну 0.1333 с суглоб має згинатися-розгинатися на кут, що надійшов з порту (усього 7 значень кута за цикл). S-PS блок конвертує безрозмірні значення з q_{Ref} (фактично – градуси) у фізичні значення в радіанах.

На рисунку 2.7 зображений наступний фрагмент моделі правої ноги робота Darth EvadeR.

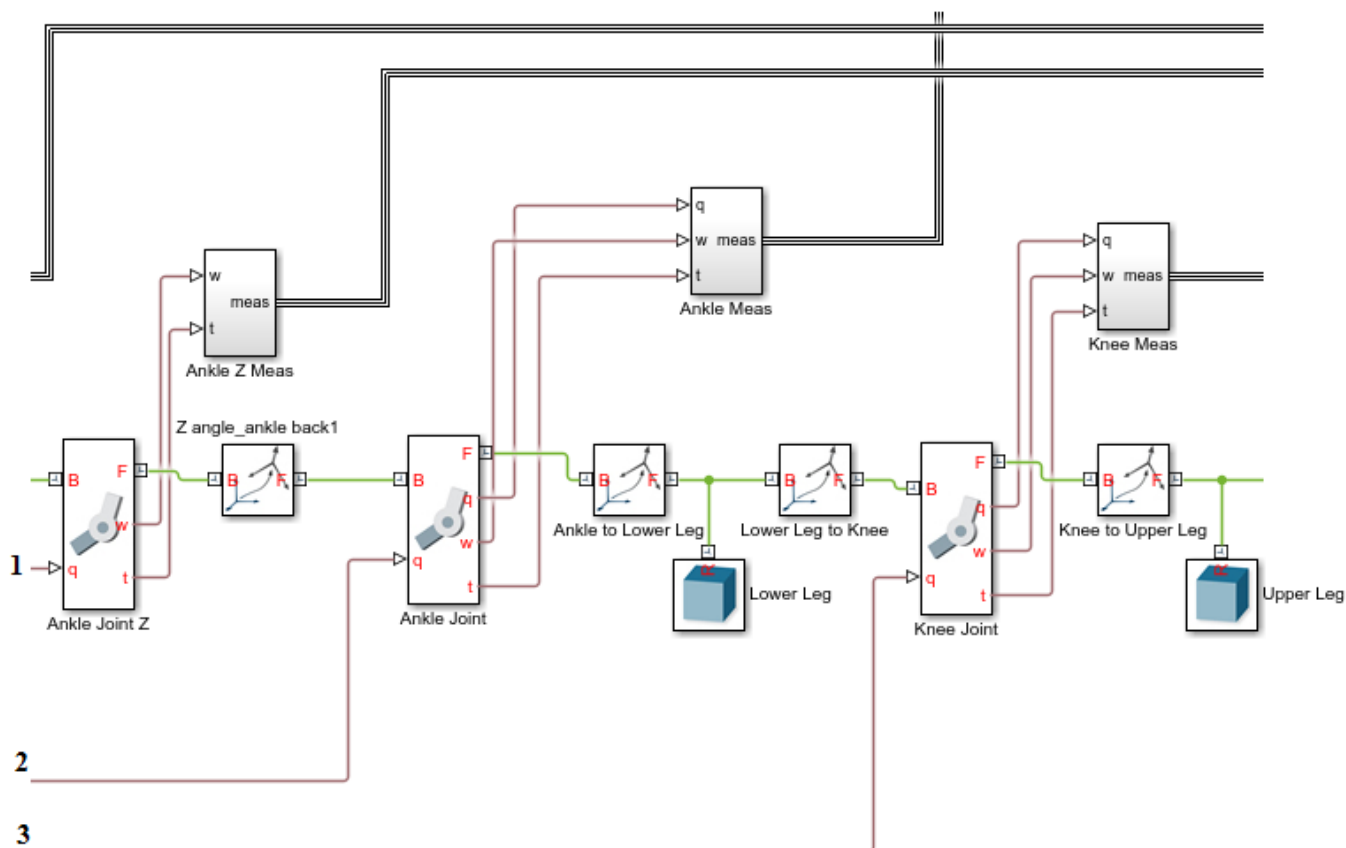


Рисунок 2.7 – Другий фрагмент моделі правої ноги робота Darth EvadeR

Фрагмент рисунку 2.7 має багато схожих блоків, котрі були розглянуті вище. Ankle Joint Z відображає суглоб, котрий дозволяє стопі здійснювати обертання у площині Z, Ankle Joint – у площині X. Lower Leg позначає безпосередньо гомілку, а Upper Leg – стегно. Knee Joint – колінний суглоб, забезпечує згинання-розгинання (обертання) коліна у площині X.

Рисунок 2.8 демонструє завершальний фрагмент будови правої ноги робота.

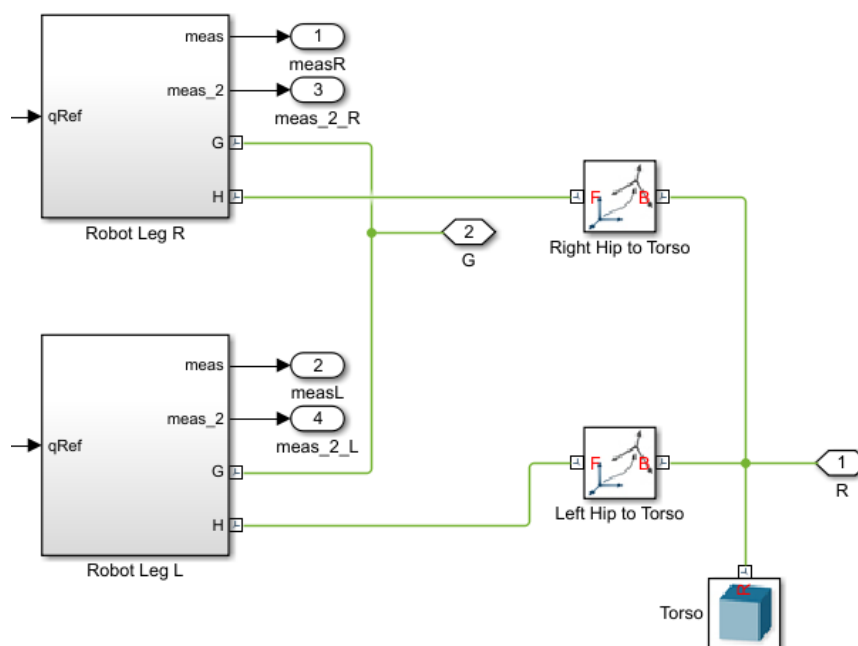


Рисунок 2.9 – Фрагмент моделі, де ноги робота з'єднуються з тулубом

Порти G та R визначають взаємодію ніг та торсу з поверхнею, котрою рухається робот. Більше інформації щодо цього моменту буде надано під час розгляду рисунків 2.11, 2.12.

Ліворуч підсистем ніг зображені входи qRef, через які відбувається подача необхідних кутів для відхилення тих чи інших частин ноги. За ними йде реалізований за допомогою блоків Simulink алгоритм уникнення роботом перешкод, котрий буде розглянуто у розділі 6, а наразі цей момент буде опущено.

Одразу перейдемо до блоків подачі кутових значень, котрі зображені на рисунку 2.10. Вони структурно поділені на 3 частини: Walking Forward, Side Walking, Grouping. Перша структура відповідає за рух робота прямолінійно – подаються кути з тією частотою та з такими значеннями, котрі забезпечують нормальну ходьбу робота вздовж вісі X. Side Walking відповідає за подачу кутів, які забезпечують роботу можливість обходити перешкоди – він рухається праворуч за допомогою додаткових степенів свободи. Формация Grouping відповідає за групування робота (триває 0.5 с) – дія, необхідна для стабілізації робота під час змінення ходьби Side Walking на Walking Forward та навпаки.

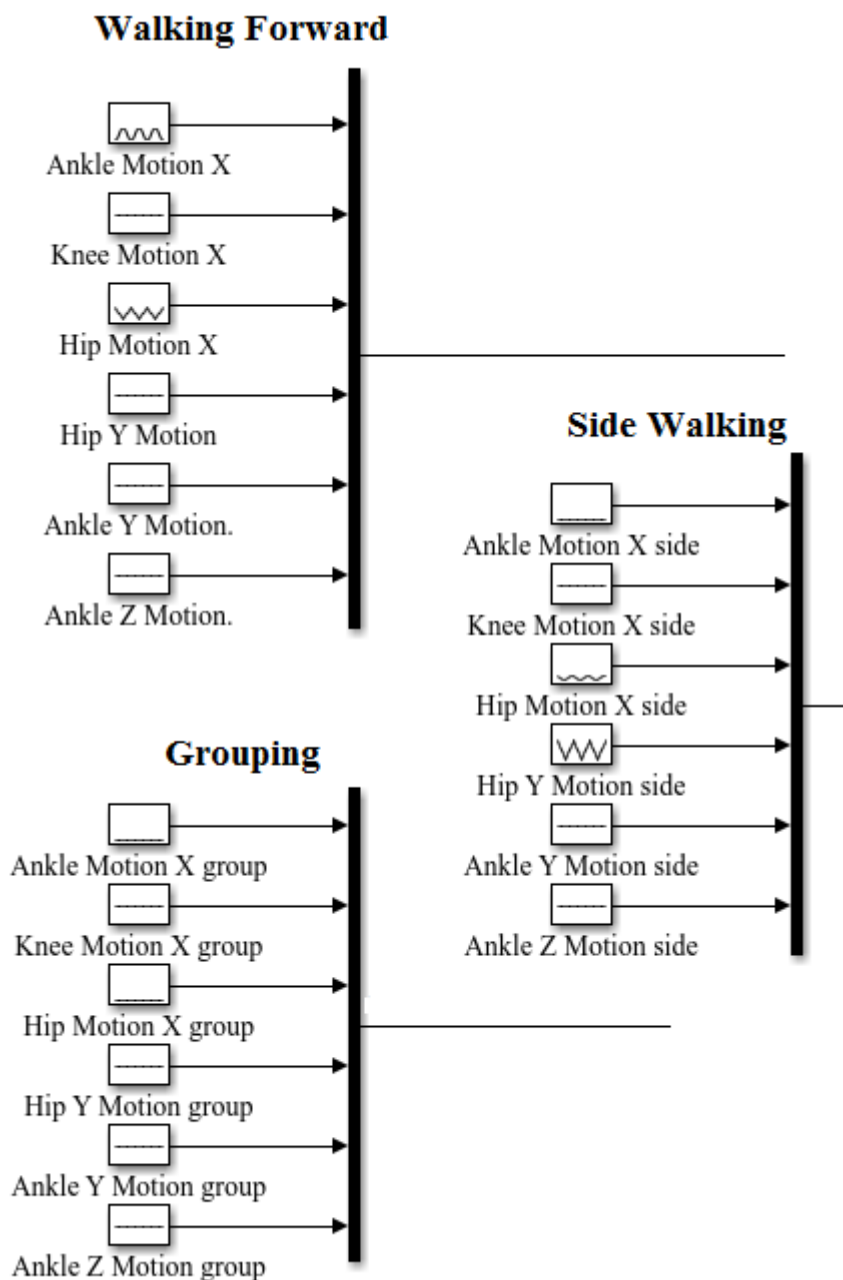


Рисунок 2.10 – Блоки подачі значень кутів

Відповідно до рисунку 2.10, блоки Ankle Motion X, Knee Motion X, Hip Motion X подають кути, що забезпечують обертання стопи, гомілки та стегна віссю X. Hip Y Motion відповідає за постачання кутів, котрі спричиняють відхилення стегна відносно осі Y, Ankle Y Motion та Ankle Z Motion – відхилення стоп віссю Y та Z.

Фактично, робот «присідає» на 0.5 с в колінах, стегнах та стопах, чим забезпечується невелике вікно спокою між різними стилями ходьби. Більш детально

дані стилі ходьби будуть розібрані у розділі 6, де пояснюється алгоритм обходження роботом перешкоди.

На рисунку 2.11 зображено, яким чином Darth EvadeR прив'язаний до навколишнього середовища моделі (перший фрагмент).

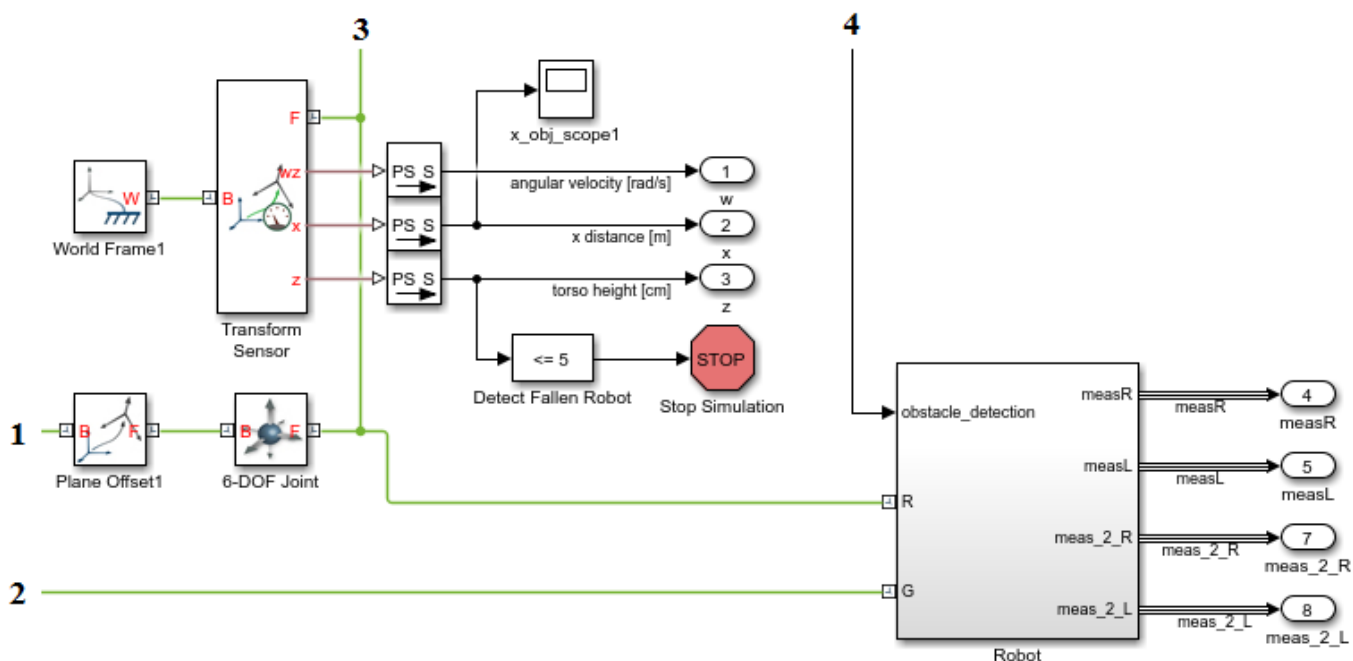


Рисунок 2.11 – Ділянка моделі, яка визначає взаємодію робота з навколишнім середовищем

Тут наявна підсистема Robot (попередньо розглянуті блоки робота інкапсульовані у цій підсистемі) та виведені у відповідні порти зчитані дані (праворуч рисунку). Блок Transform Sensor дозволяє вимірювати залежні від часу зв'язки між двома кадрами, у даному випадку – між роботом та його оточенням (перешкода та поверхня ходьби). Наприклад, він дозволяє знаходити висоту торсу над поверхнею, пройдену роботом дистанцію та його кутову швидкість у кожен фіксований момент часу.

Detect Fallen Robot дозволяє зупиняти моделювання у разі, якщо робот впаде. Стрілки, що позначені як 3 та 4 ведуть до блоків, котрі дозволяють визначити відстань від робота до перешкоди (розглядається детальніше при описі рисунку

2.13). Стрілки 1 та 2 ведуть безпосередньо до поверхні та перешкоди; перший шлях (де 6-DoF Joint) дозволяє роботу вільно рухатися у всіх трьох координатах простору відносно оточення, другий просто встановлює зв'язок між оточенням та роботом.

На рисунку 2.12 другий фрагмент ділянки, котрий визначає площину ходьби, її просторові налаштування, а також просторові налаштування перешкоди.

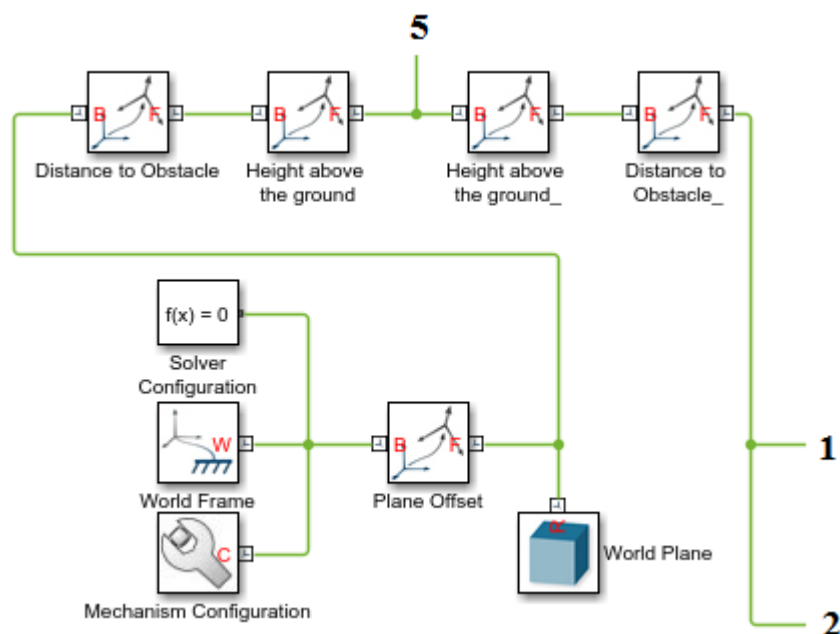


Рисунок 2.12 – Блоки, котрі визначають площину, якою рухається робот, та просторові налаштування перешкоди

Відповідно до рисунку 2.12: блоки Solver Configuration, World Frame, Mechanism Configuration визначають базові налаштування усієї моделі і використовуються в усіх моделях, що створюються за допомогою бібліотеки Simscape Multibody. Plane Offset визначає розташування та певні параметри площини, якою пересувається робот.

Distance to Obstacle визначає розташування блоку перешкоди відносно робота, Height above the ground відповідає за висоту перешкоди над площиною ходьби, інші два блоки відповідно «закривають» зону дії параметрів цих блоків Distance to Obstacle та Height above the ground.

Рисунок 2.13 визначає перешкоду та механізм її фіксування роботом.

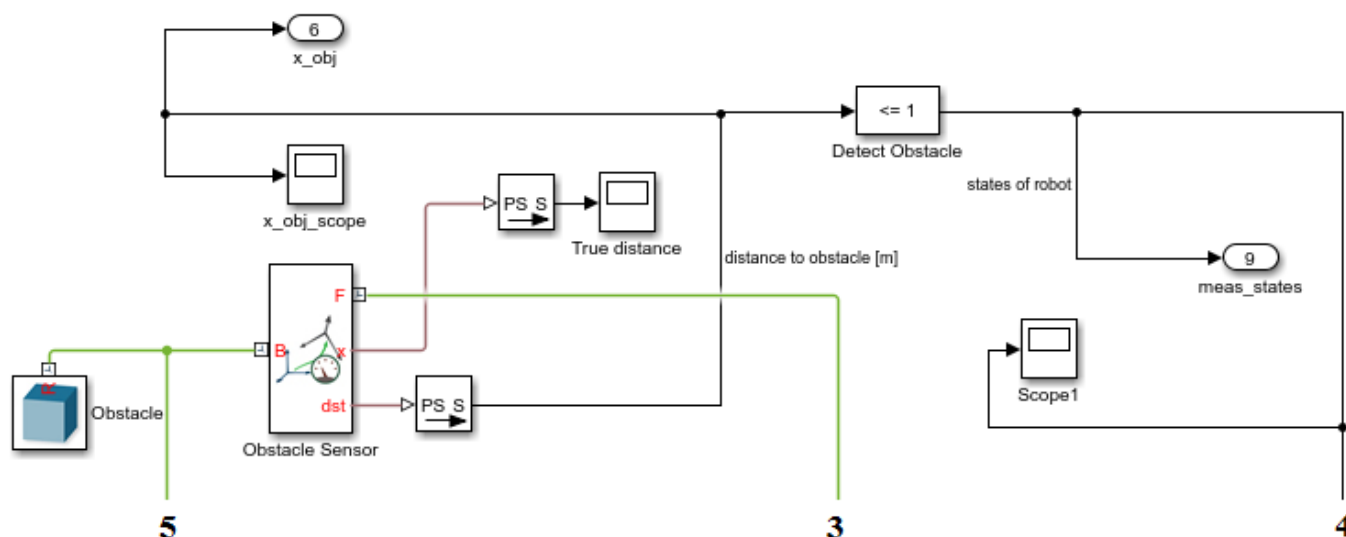


Рисунок 2.13 – Фрагмент схеми, котра відповідає за відображення та взаємодію з перешкодою

Відповідно до рисунку 2.13: Obstacle визначає розміри перешкоди та її присутність загалом. Obstacle Sensor – детектор, котрий виходом B пов'язаний з перешкодою, а виходом F – із самим роботом, таким чином встановлюючи між ними зв'язок, котрий дозволяє відстежувати необхідний параметр зближення з перешкодою. Вихід x визначає відстань між Darth EvadeR`ом та завадою. Як тільки робот зблизиться з перешкодою до відстані 1 м, блок Detect Obstacle зафіксує цю подію і подасть на вихід 4 логічну одиницю, що буде використано у алгоритмі обходження завади.

Висновки до розділу 2

Таким чином, у даному розділі були досягнуті наступні важливі цілі:

1. Описано динаміку антропоморфних механізмів із сімома степенями свободи. Наведено рівняння, що описує динаміку як одноопорного руху (ходьба людини), так і двоопорного. Відомості, розкриті у даному розділі,

допомагають зрозуміти математичне підґрунття усіх двоногих механізмів, включаючи двоногого крокуючого робота, котрий досліджується в даній магістерській дисертації.

2. Розроблено дві структурні схеми: перша демонструє основні структурні компоненти готової авторської моделі робота, а друга – модернізованої моделі робота. Вони дозволяють розуміти той фундамент, котрий закладений до кожної моделі, навіть без представлення більш детальних та інформативних пояснень. Модернізований варіант схеми був розроблений на основі авторської моделі робота, створеної одним з популяризаторів MATLAB-продукції Себастьяном Кастро. Усі посилання на його розробки були залишені у розділі відповідно до чинних вимог авторських прав.
3. Детально описано схему ДКР Darth EvadeR, виконану в інтерактивному середовищі Simulink. Оригінальна схема [35, 36] була модернізована таким чином, аби робот зміг розпізнавати перешкоду та обходити її відповідно до алгоритму, що буде наведено у наступному розділі.

3 ОПИС ЗАПРОПОНОВАНИХ МОДЕРНІЗАЦІЙ МОДЕЛІ РОБОТА DARTH EVADER

3.1 Характеристики авторського двоногого робота

Авторський робот, відповідно до заяв самого автора, був розроблений спеціально для того, щоб познайомити аудиторію ПЗ MATLAB з бібліотекою Simscape Multibody, призначеною для моделювання різних механічних 3D систем, і показати, на що вона здатна. Крім того, усі файли розробленої моделі були викладені на сервісі GitHub для того, аби кожен бажаючий міг завантажити її собі та використовувати у власних цілях для проведення різноманітних експериментів над моделлю робота [36].

Таким чином, після ретельного дослідження авторської моделі були виявлені наступні її характеристики:

- вага: 0.6273 кг (вага обрахована самостійно, оскільки автор у відповідних блоках, що позначали частини тіла, оперував не вагою, а щільністю);
- висота: 28 см;
- ширина: 10 см;
- довжина: 5 см;
- кількість степенів свободи: 6 (2 для гомілковостопних суглобів, 2 для колінних, 2 для стегнових – усі здійснюють обертання лише відносно осі X).

ДКР здатний відтворювати рухи нижніх кінцівок подібно до людини і здійснювати прямолінійну ходьбу в одну сторону (ходьба зображена на рисунку 3.1). Цей набір характеристик більше нагадує певний ляльковий прототип справжнього робота.

Однак, модель працює – робот ходить і успішно долає різні відстані. Це дає широкий спектр можливостей щодо вдосконалення робота та проведення різного роду дослідів та експериментів.

У розділі 1 була визначена проблематика сучасних ДКР – це великі грошові та часові затрати на виготовлення роботів. Щоб зменшити витрати цих ресурсів, доцільно вдосконалити авторську модель шляхом наближення її до людських параметрів і шляхом розробки алгоритму уникнення перешкод, котрий буде достатньо простим і не вибагливим до апаратного та програмного забезпечення. Окрім цього, слід приділити особливу увагу підвищенню загальної ефективності динаміки крокуючого робота шляхом оптимізації його прямолінійної ходьби.

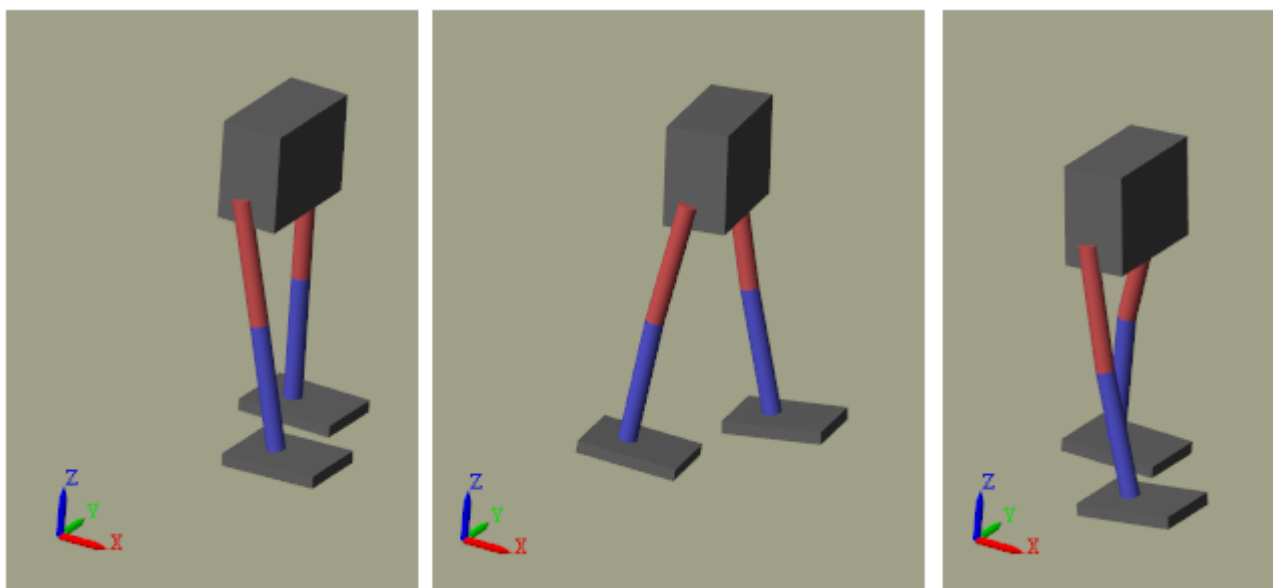


Рисунок 3.1 – Ходьба авторського ДКР

3.2 Шляхи та результати модернізації моделі Darth EvadeR

Відповідно до вище наведених цілей модернізації робота, а також зі врахуванням відомостей про існуючих роботів, що наведені у розділі 1, виконані наступні кроки [37]:

1. Перевизначення вагової системи блоків, котрі позначають частини тіла робота. Авторська модель в принципі не мала параметру «вага» - його альтернативою була щільність, а тому напругу оперувати вагою та визначати її не можна було, що приносило значні незручності та додавало зайвої складності моделі.

2. Зміна ваги робота на таку, що більш наближена до людини. Модернізована модель робота важить 172 кг: по 60 кг на стопи, по 8 кг на гомілки та на стегна, 20 кг на тулуб. Такі великі значення для стоп були обрані через те, що двоногі роботи досить важко врівноважуються у порівнянні з екземплярами, котрі мають більшу кількість кінцівок [38]. Тому, якщо у простого робота, який не має додаткових приводів чи інших механізмів перерозподілу центру маси, вся вага сконцентрована у тулубі (при цьому елементи ніг мають відносно малу вагу), його переміщення має більшу інерційність, а сам робот стає дуже складним у врівноваженні, і це різко підвищує ризик падіння та так званих «заносів». Оскільки Darth EvadeR є простою моделлю робота, він має примітивні способи впливу на проблему рівноваги і найбільш простий з них – обважування стоп, котрі найближче за всі частини тіла знаходяться до площини ходьби. При розробці реального прототипу цей момент легко обігрується, наприклад, гідравлічними або іншими механізмами, а також спеціально налаштованими контролерами, що здатні оперативно регулювати баланс робота в режимі реального часу.

3. Зміна розмірів моделі робота на більш наближені до людини. Розміри складових модернізованої моделі робота (висота, вісь Z): стегна — 38 см, гомілки — 40 см, торс — 32 см. Вказані розміри стегон та гомілок мають такі ж розміри, як у людини зростом 1 м 75 см.

4. Зміна параметрів взаємодії моделі робота з поверхнею (вище згадувані контактні сили) таким чином, аби вона максимально відповідала реальності. Досягнуто експериментальним шляхом.

5. Впровадження додаткової мобільності моделі робота шляхом додавання нових 6-ти степенів свободи. Модернізована модель здатна забезпечувати: відхилення стегон відносно осі Y (рух праворуч-ліворуч – рискання), відхилення стоп відносно осей Y та Z (рискання та крен, відповідно). Darth EvadeR має 12 степенів свободи. Отже, у даної моделі більш наближена до людської кількість степенів свободи [39].

6. Реалізація скриптів, які дозволяють відстежувати швидкість робота у кожен момент часу, підраховувати його середню швидкість, а також обчислювати енергоспоживання робота за весь час моделювання.

7. Додавання перешкоди на шляху робота. Вона виглядає як невеликий паралелепіпед і має такі розміри: 40 см – довжина, 80 см – ширина, 16 см - висота.

8. Реалізація виявлення роботом завади та її уникнення з подальшим відновленням прямолінійного руху (направлене вздовж осі X).

Darth EvadeR має наступний вигляд (рисунок 3.2):

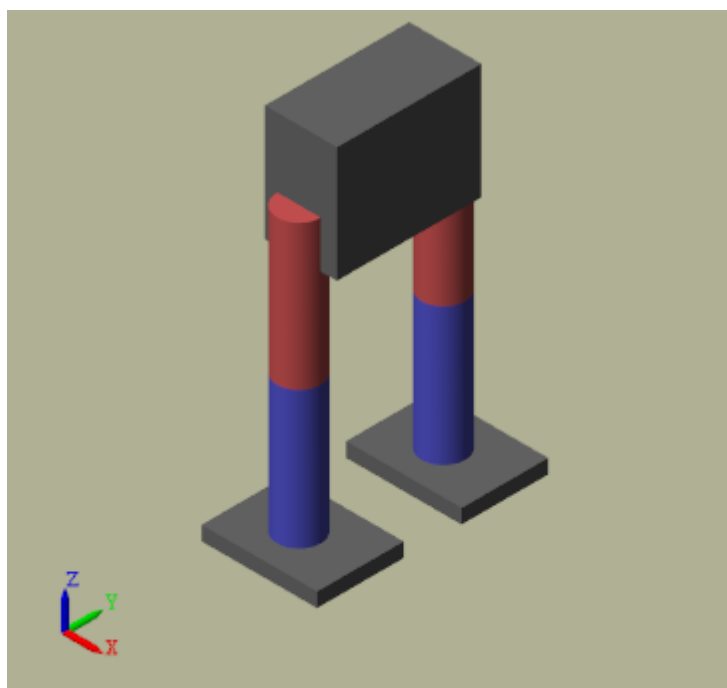


Рисунок 3.2 – Зовнішній вигляд моделі Darth EvadeR

Також, був змінений стиль самої прямолінійної ходьби за рахунок введення інших кутів нахилу у стегнах, колінах та стопах, аби він був більш стабільним та сприяв врівноваженню моделі і не провокував падіння Darth EvadeR`а.

3.3 Опис концепції обходження перешкоди роботом

Перед безпосередньою розробкою блок-схеми алгоритму доцільно навести загальні відомості щодо ходьби та обходження Darth EvadeR`ом перешкоди на шляху.

Отже, з минулих розділів відомо, що цикл ходьби робота складає 0.8 с, тобто переставлення однієї ноги займає приблизно 0.4 с. За 0.8 с кожен суглоб отримує по 7 значень кутів, на які він має здійснити переміщення.

В ході моделювання робот виконує наступні дії:

1. Запуск моделі. Робот на 0.5 с стає в позицію Grouping (групування). Ця позиція характеризується синхронно зігнутими стопами, колінами та стегнами (-15° , 30° , -30° відповідно) – вона допомагає Darth EvadeR`у зайняти досить стійку позицію, вийшовши з якої у робота буде менше всього шансів втратити рівновагу. Побачити цю позу можна на рисунку 3.3.

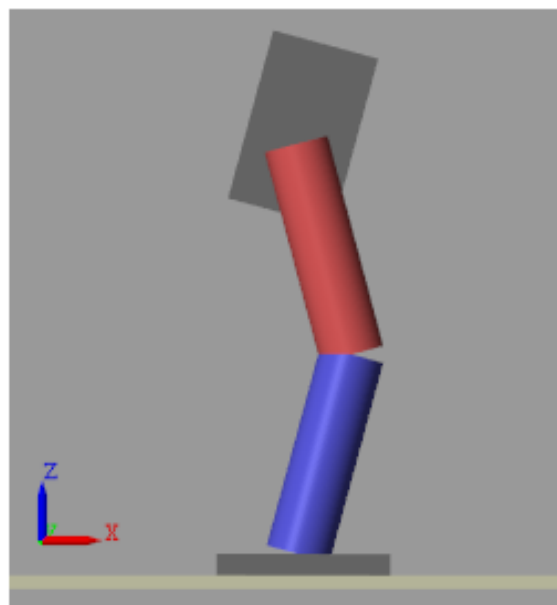
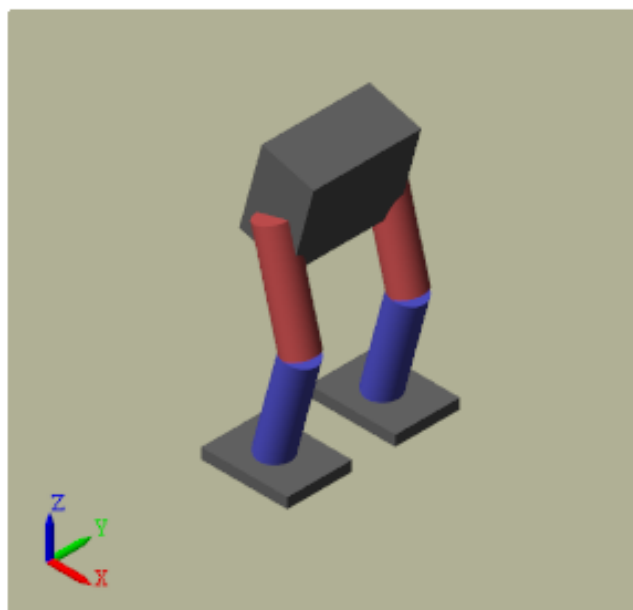


Рисунок 3.3 – Робот у позиції Grouping

2. По завершенню 0.5 с робот переходить у стан Walking Forward. Цей стан характеризується циклами ходьби по 0.8 с, під час яких робот рівномірно

наближається до перешкоди, проходячи 1.5 м. Робот у цьому стані зображений на рисунку 3.4.

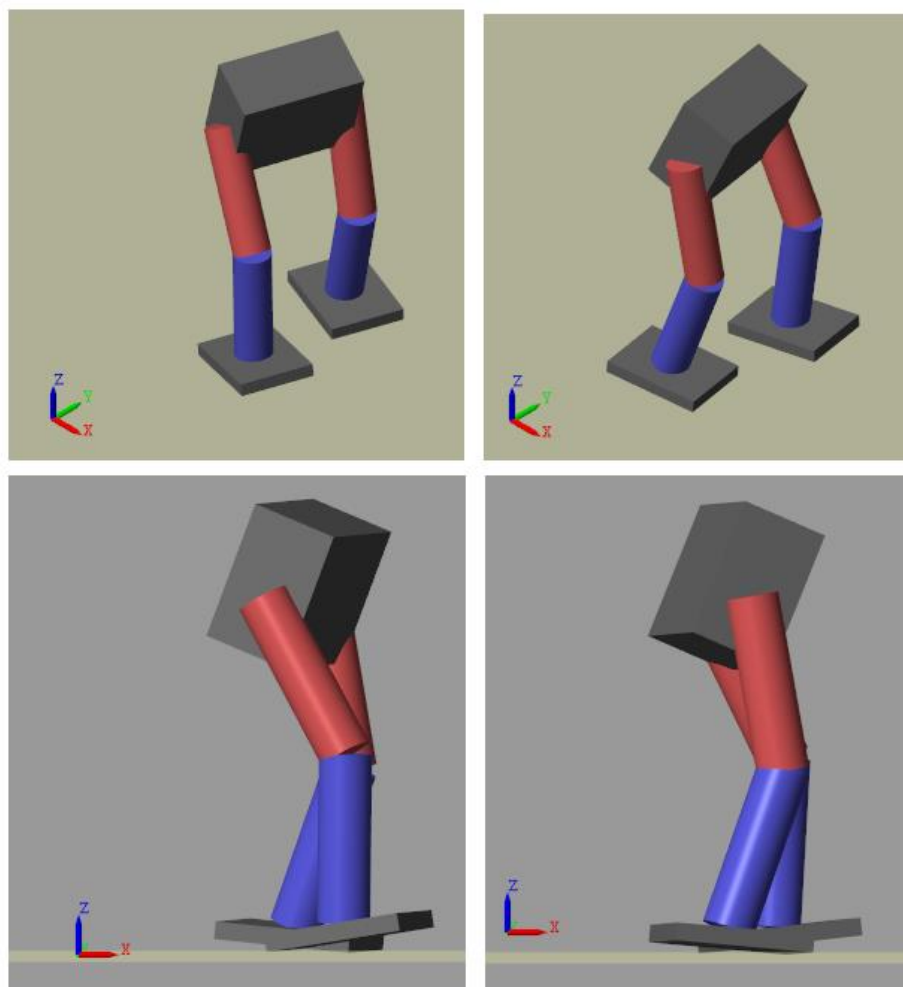


Рисунок 3.4 – Darth EvadeR у стані прямолінійної ходьби

3. Коли Obstacle Sensor фіксує зближення робота з перешкодою (відстань між ними 1 м), робот знову переходить до стану Grouping тривалістю 0.5 с для того, щоб взяти паузу та врівноважитися перед початком руху вбік (рисунок 6.3). Перешкода має наступні параметри: ширина – 40 см, довжина – 80 см, висота – 16 см. Робот у вказаному стані зображений на рисунку 3.5.

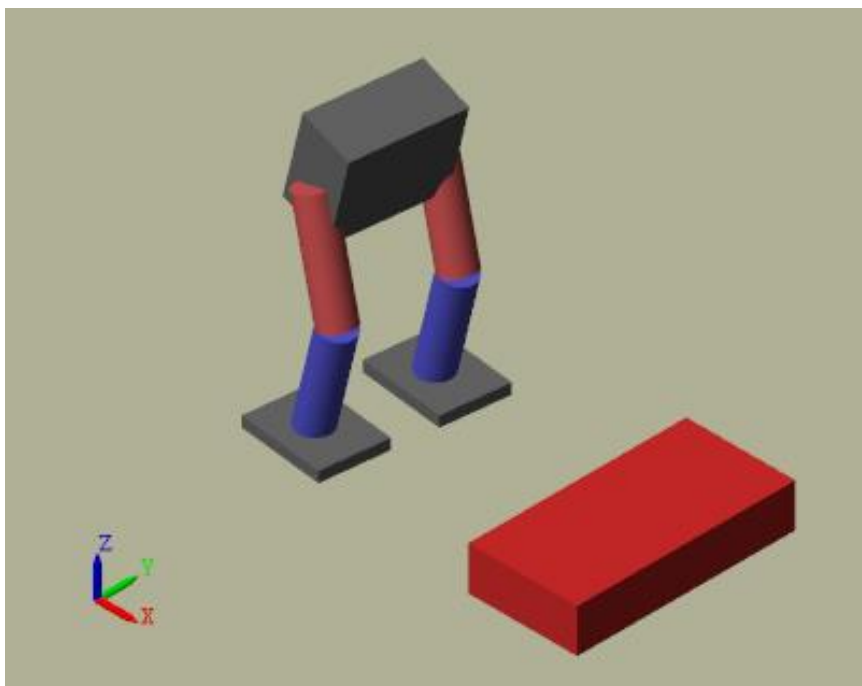


Рисунок 3.5 – Робот у стані Grouping перед перешкодою

4. По завершенню відведеного на Grouping проміжку часу робот переходить до стану Side Walking і розпочинає рух ліворуч, використовуючи можливості розширеної кількості степенів свободи. Таким чином він збільшує дистанцію між собою та перешкодою. Спосіб обходження завади зображено на рисунку 3.6.
5. Після того, як робот покинув діапазон перешкоди (1 м), він покидає стан Side Walking і переходить в режим Walking Forward, відновлюючи прямолінійний початковий рух. Відновлену ходьбу можна побачити на рисунку 3.7.

Зупинка моделювання робота відбувається у трьох випадках:

- робот впав;
- завершився час моделювання;
- ручна зупинка моделювання.

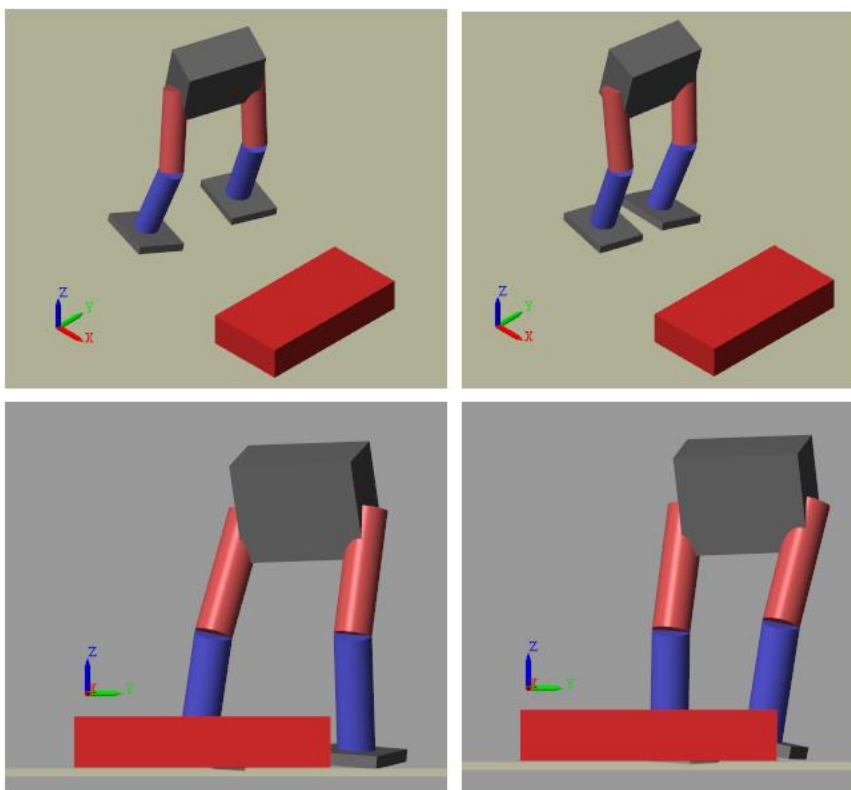


Рисунок 3.6 – Переміщення робота ліворуч (стан Side Walking)

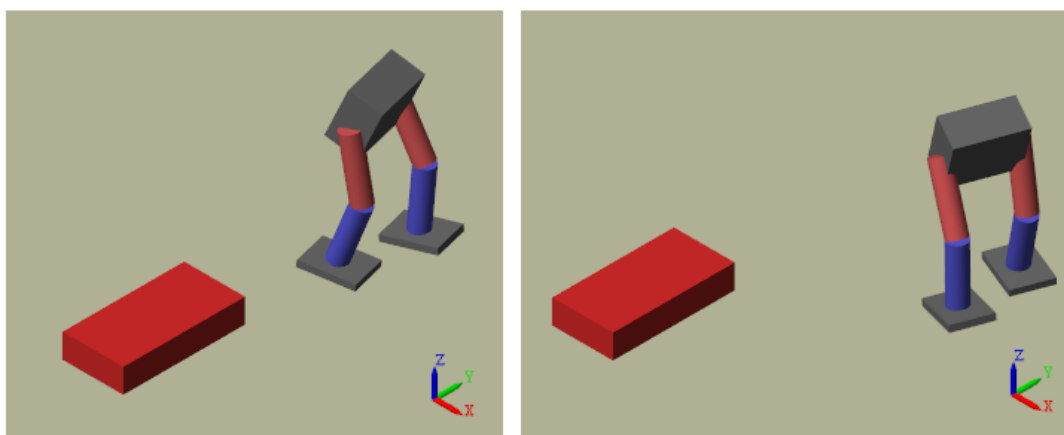


Рисунок 3.7 – Відновлення прямолінійного початкового руху

Таблиці 3.1 – 3.3 відображають значення кутів, котрі подаються на суглоби робота Revolute Joints протягом одного циклу ходьби. Таблиця 3.1 демонструє значення для режиму Grouping, таблиця 3.2 для Walking Forward, таблиця 3.3 для Side Walking.

Таблиця 3.1 – Кути, що подаються на суглоби робота у стані Walking Forward

Назва суглобів	Кортеж значень (градуси)
Гомілково-стопні (ротація віссю X)	-15, -15, -10, -10, -10, -15, -15
Гомілково-стопні (вісь Y)	0, 0, 0, 0, 0, 0, 0
Гомілково-стопні (вісь Z)	0, 0, 0, 0, 0, 0, 0
Колінні (вісь X)	30, 30, 30, 30, 30, 30, 30
Стегнові (вісь X)	-30, -38, -46, -52, -46, -36, -30
Стегнові (вісь Y)	0, 0, 0, 0, 0, 0, 0

Таблиця 3.2 – Кути, що подаються на суглоби робота у стані Grouping

Назва суглобів	Кортеж значень (градуси)
Гомілково-стопні (ротація віссю X)	-15, -15, -15, -15, -15, -15, -15
Гомілково-стопні (вісь Y)	0, 0, 0, 0, 0, 0, 0
Гомілково-стопні (вісь Z)	0, 0, 0, 0, 0, 0, 0
Колінні (вісь X)	30, 30, 30, 30, 30, 30, 30
Стегнові (вісь X)	-30, -30, -30, -30, -30, -30, -30
Стегнові (вісь Y)	0, 0, 0, 0, 0, 0, 0

Таблиця 3.3 – Кути, що подаються на суглоби робота у стані Side Walk

Назва суглобів	Кортеж значень (градуси)
Гомілково-стопні (ротація віссю X)	-15, -15, -15, -15, -15, -15, -15
Гомілково-стопні (вісь Y)	0, 0, 0, 0, 0, 0, 0
Гомілково-стопні (вісь Z)	4, 4, 4, 4, 4, 4, 4
Колінні (вісь X)	30, 30, 30, 30, 30, 30, 30
Стегнові (вісь X)	-30, -32, -35, -35, -32, -30, -30
Стегнові (вісь Y)	-5, -8, -12, -16, -13, -8, -6

Як свідчать покази таблиць 3.1 – 3.3, в даній моделі гомілково-стопний суглоб Y застосовувати не довелося.

3.4 Розроблення блок-схеми алгоритму функціонування моделі робота

Тепер наведемо загальний алгоритм роботи моделі Darth EvadeR у вигляді блок-схеми. Її можна побачити на рисунку 3.8.

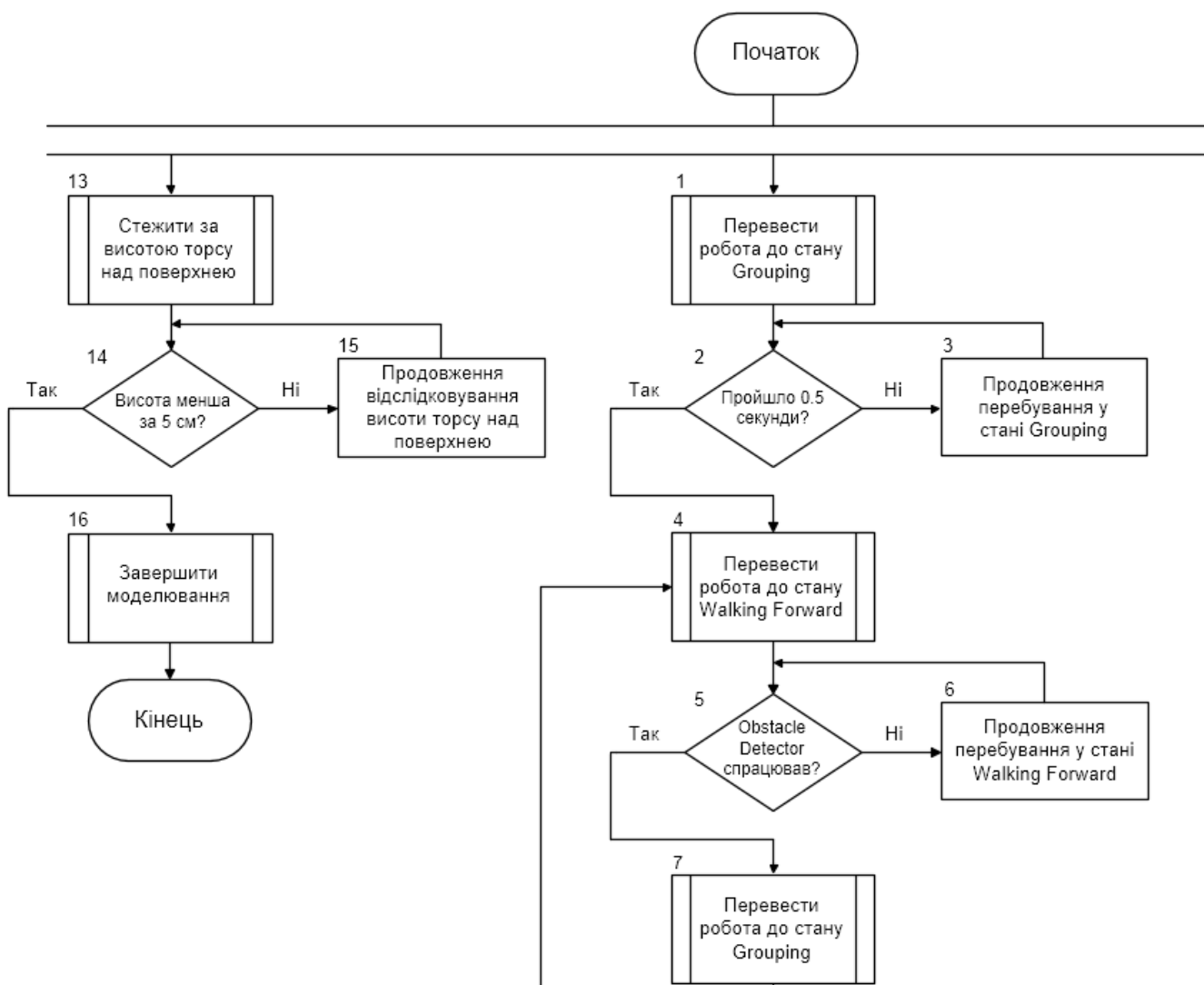


Рисунок 3.8 – Перша частина блок-схеми алгоритму відпрацювання моделі робота Darth EvadeR

На рисунку 3.9 можна побачити продовження блок-схеми.

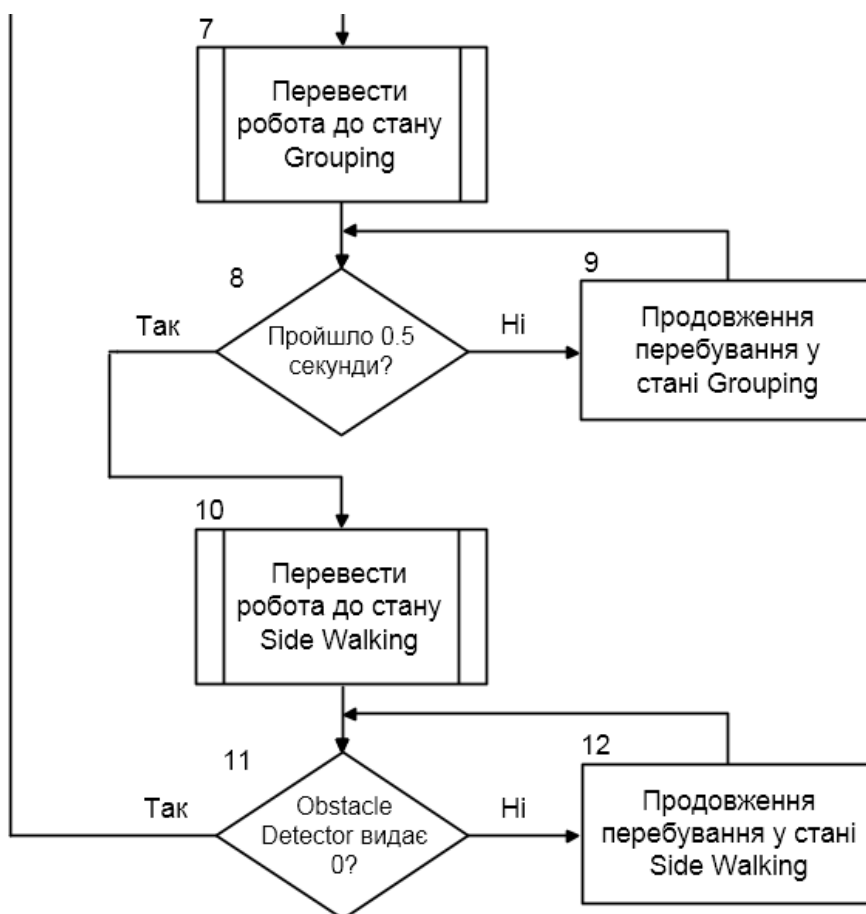


Рисунок 3.9 - Продовження блок-схеми алгоритму відпрацювання моделі робота Darth EvadeR

З рисунків 3.8 та 3.9 легко бачити, що опис діяльності моделі робота у підрозділі 3.1 повністю співпадає з алгоритмом, відображеним у наведеній блок-схемі.

3.5 Реалізація за допомогою засобів Simulink логіки переміщення робота при виявленні перешкоди

Логіка переміщення робота при виявленні перешкоди реалізована за допомогою базових блоків інтерактивного середовища моделювання MATLAB/Simulink. В основному вона побудована на блоках Switch, котрі перемикають подачу тих чи інших значень кутів на суглоби Revolute Joints. Також

використовуються блоки затримки (Delay) та констант (Constant) для утворення певних інтервалів між станами робота та їх підтримки. Ознайомитися з реалізованою логікою переміщення робота можна за допомогою рисунку 3.10.

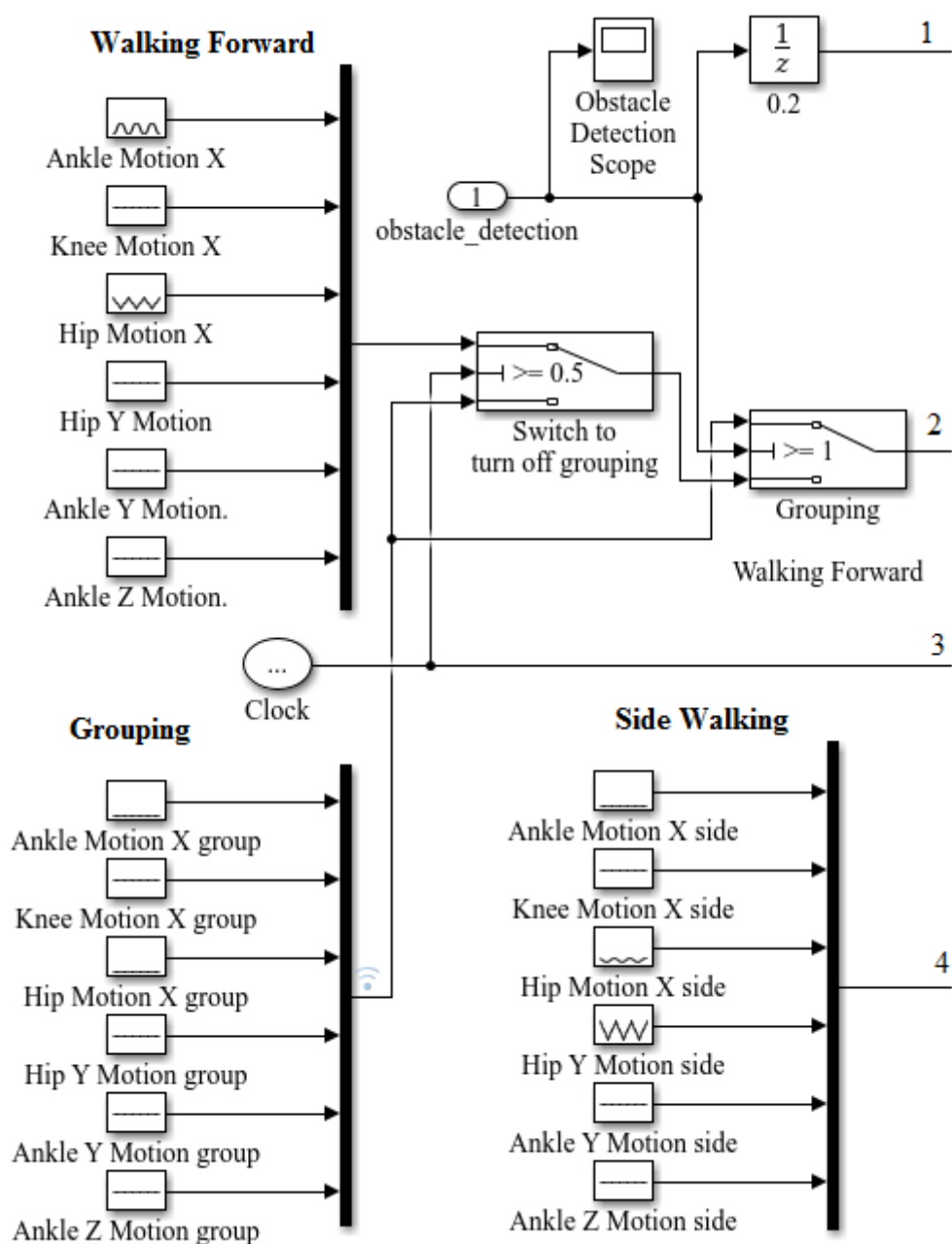


Рисунок 3.10 – Перший фрагмент реалізації логіки обходження завади роботом у Simulink [37]

Продовження реалізації логіки на рисунку 3.11.

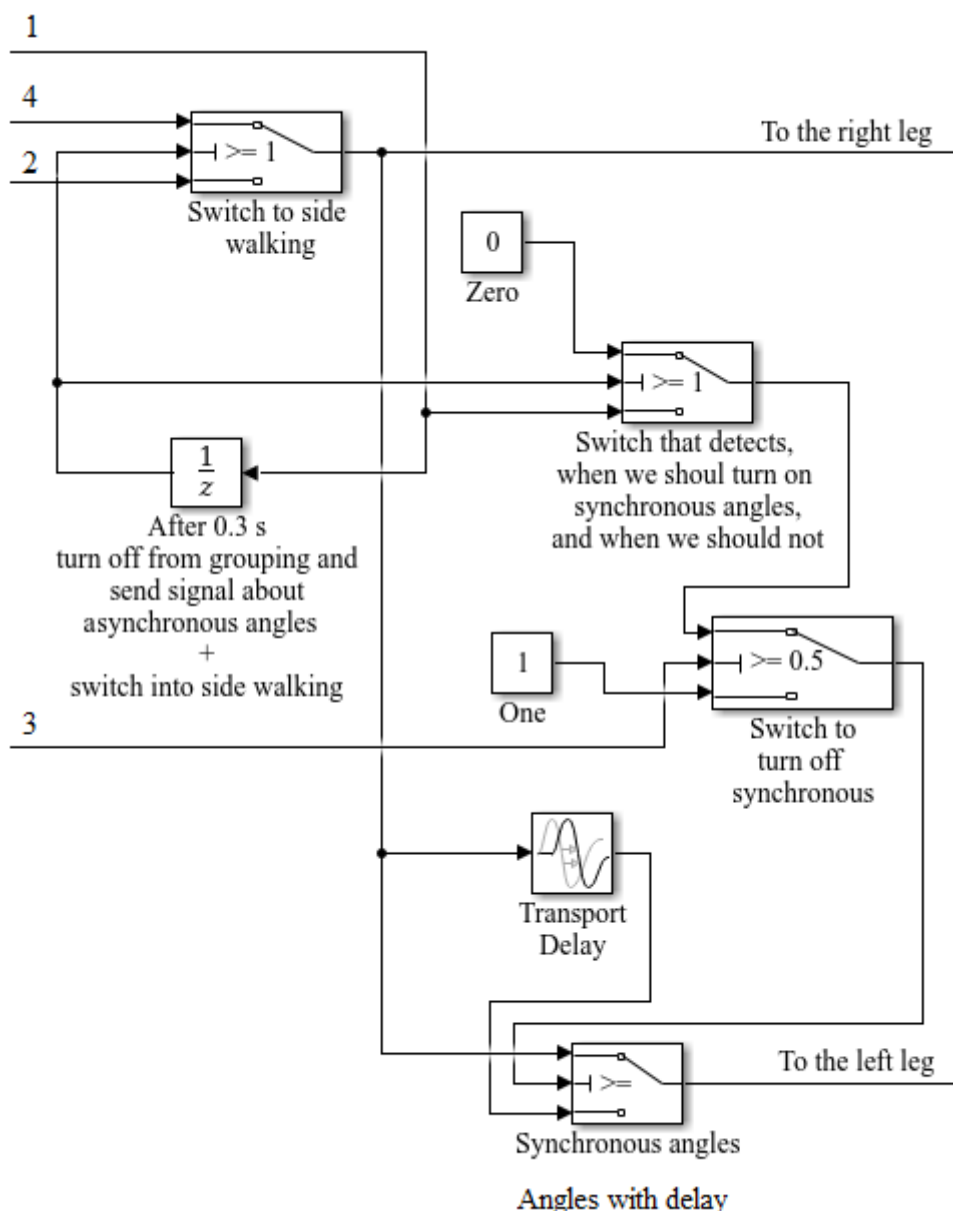


Рисунок 3.11 – Другий фрагмент реалізації логіки обходження завади роботом у Simulink [37]

З рисунків 3.10 та 3.11: усі блоки зі схеми підписані відповідно до функцій, котрі вони виконують. Наведемо додаткові пояснення до рисунку 3.10:

- блоки під заголовком Walking Forward, Grouping та Side Walking розглядалися при описі рисунку 2.10 у розділі 2 і відповідають за режими ходьби робота;

- Clock – блок, котрий підраховує час моделювання та використовується для забезпечення перебування робота у стані Grouping під час перших 0.5 с моделювання;
- Switch to turn off grouping – перемикає режим ходьби з Grouping на Walking Forward після того, як Clock зафіксує час, рівний 0.5 с; спрацьовує тільки раз на початку моделювання, а після 0.5 с завжди на вихід подає режим Walking Forward;
- obstacle_detection – вхідний порт, котрий подає логічну одиницю, коли робот зближується з перешкодою (1 м відстань) та логічний нуль при відстані між роботом і завадою більшою за 1 м;
- Obstacle Detection Scope – дозволяє виводити на графік моменти фіксування зближення робота з перешкодою;
- Grouping / Walking Forward – блок, котрий відповідає за переключення між режимом Grouping та Walking Forward і спрацьовує всякий раз, коли робот зближається з перешкодою;
- блок затримки, позначений як «0.2» – значення затримки у секундах, дозволяє відтермінувати надходження сигналу про фіксацію перешкоди на шляху, аби робот встиг перейти спочатку до стану Grouping для групування перед початком процесу обходження перешкоди;

Пояснення до рисунку 3.11:

- Switch to side walking – перемикач на ходьбу в режимі Side Walking;
- Synchronous angles / Angles with delay – блок, який відповідає за синхронізацію лівої ноги робота; в режимах Walking Forward та Side Walking ліва нога робота робить запізнення у розмірі 0.4 с, що дозволяє переставляти ноги по чергові, а у режимі Grouping всі рухи відбуваються повністю синхронно без затримок;
- Transport Delay – блок, що вносить затримку до руху лівої ноги у розмірі 0.4 с;
- Switch to turn off synchronous – перемикач, який відповідає за вимкнення режиму синхронності робота після 0.5 с з початку роботи Darth EvadeR`a,

приймає булеві значення і призначений для синхронізації рухів робота перші 0.5 с у стані Grouping, а надалі просто передає значення з виходу блоку «Switch that detects...» до блоку Synchronous angles / Angles with delay;

- блок під назвою «Switch that detects...» відповідає за визначення того, коли необхідно подавати сигнал на Switch to turn off synchronous для ввімкнення/вимкнення синхронності ніг; подає відповідний сигнал про синхронність через 0.2 с після детектування зближення з перешкодою;
- блок затримки під назвою «After 0.3 s ...» - вносить затримку 0.3 с після отримання сигналу про фіксування перешкоди; після цього подається сигнал на ввімкнення асинхронних кутів для того, щоб робот міг вийти із синхронного стану Grouping і рухатися асинхронно в режимі Side Walking.

Також, варто відмітити, що блок «After 0.3 s ...» та блок затримки «0.2» забезпечують плавний та надійний перехід робота з режиму Walking Forward до Grouping та з Grouping до Side Walking.

3.6 Програмування визначення середньої швидкості моделі робота протягом усього часу моделювання

Аби визначити середню швидкість моделі робота, у даній магістерській дисертації будуть застосовані скрипти. Вони дозволять у зручному вигляді виконувати обробку інформації, котра отримується після відпрацювання моделі, і робити оцінку необхідних параметрів робота. Увесь програмний код був написаний за допомогою мови програмування MATLAB, яка вбудована до однойменного ПЗ.

Швидкість є важливим параметром для будь-якого робота не тільки як самостійна величина, а також, як і величина визначення ефективності механізму. Детальніше про це буде йти мова у розділі 4.

Основна відмінність скрипту визначення швидкості для моделі Darth EvadeR`а відносно інших роботів полягає у тому, що заміри швидкості мають виконуватися протягом усього часу виконання Darth EvadeR`ом свого завдання, яке полягає у

наступному: робот проходить 1.5 м і зупиняється перед перешкодою (датчик фіксує відстань до перешкоди, яка дорівнює 1 м), після цього він розпочинає обходити 0.4 м перешкоди і заходить за неї на таку відстань, щоб ліворуч та праворуч від робота був вільний простір. Більшу наглядність ця дія має на рисунку 3.12.

Реалізовано даний скрипт було дуже просто: із зібраних даних щодо пройденої роботом відстані у кожен момент часу обчислюється середнє значення швидкості. Результат – шукана середня швидкість робота, яку має робот під час виконання свого завдання. Виводиться графік швидкості робота у кожен момент часу, а також повідомлення про конкретну величину.

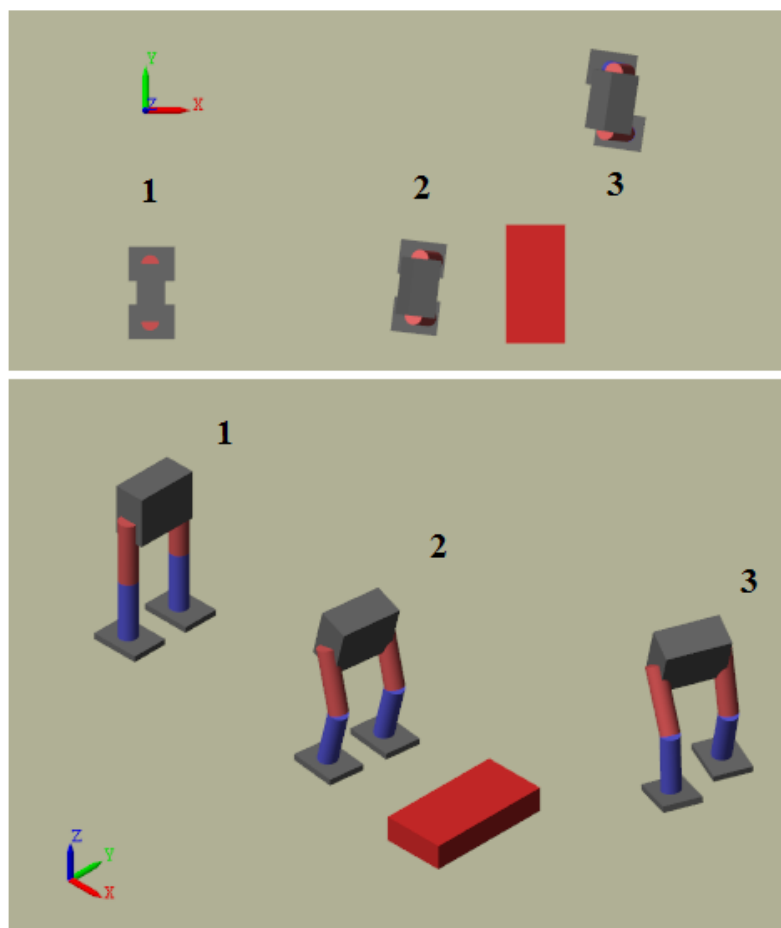


Рисунок 3.12 – Стадії, які має пройти Darth EvadeR відповідно до завдання: 1 – робот у початковій позиції, 2 – завада виявлена, тому робот у стані Grouping і починає її обходити, 3 – робот знаходиться за завадою (праворуч та ліворуч вільний простір)

На рисунку 3.13 зображено відпрацювання скрипту – графік швидкості, а також повідомлення про обраховану середню швидкість. Моделювання відбулося з параметрами, вказаними у таблицях 3.1 – 3.3. Алгоритм ходьби робота також був наведений у цьому розділі, однак рисунок 3.13 зводить пояснення до купи і яскраво демонструє основні етапи функціонування робота.

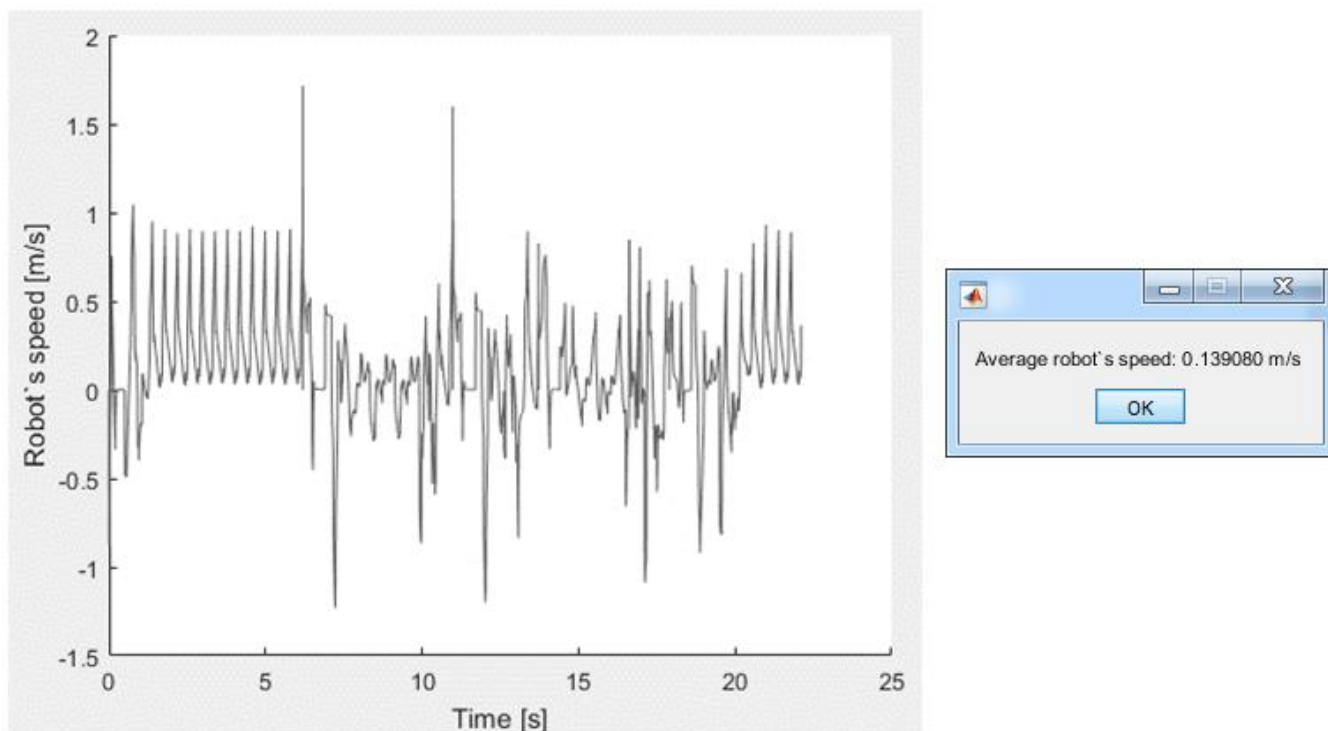


Рисунок 3.13 – Відпрацювання скрипта: графік швидкості Darth EvadeR`а та повідомлення про обраховану середню швидкість робота

З рисунку 3.13 видно: приблизно до 6-ї секунди робот йшов прямолінійно, потім зупинився (виявив перешкоду) і розпочав рухатися ліворуч; приблизно на 20.5 секунді він відновив прямолінійну ходьбу і продовжив вільний рух. Програмний код, описаний у цьому розділі, можна знайти у Додатку А.

3.7 Формула для визначення середньої потужності, котру споживає модель робота

Для визначення потужності було застосовано більш складний алгоритм. Як відомо, в електродинаміці потужність визначається як добуток сили струму та напруги на ділянці електричного кола. Визначення можна зобразити наступною формулою (3.1), підкорегувавши її під робототехнічні системи:

$$P_{\text{сер}} = I_{\text{сер}} \cdot U, \quad (3.1)$$

де $I_{\text{сер}}$ — середнє значення сили струму, що подається на приводи, А;

U — номінальна напруга на приводах, В.

Це класична формула, яка є зручною у визначенні середнього значення потужності, що подається на приводи робота. Її доцільно використовувати на вже існуючому роботі, де є можливість робити точні виміри сили струму, за допомогою якої регулюється рух кожної кінцівки.

Якщо крокуюча робототехнічна система ще не має хоча б якогось мінімального робочого прототипу, а існує лише у вигляді віртуальної моделі, потужність також можна виміряти. Для підрахунку потужності кожного приводу слід використовувати менш відому формулу через обертальний момент та кутову швидкість [40], яка виглядає наступним чином (3.2):

$$P = M \cdot \omega, \quad (3.2)$$

де M — обертальний момент, прикладений до з'єднуючого шарніру, Н*м;

ω — кутова швидкість обертання валу, с⁻¹.

Отже, при моделюванні робота у спеціалізованому програмному середовищі слід реалізувати виведення значень обертаючого моменту та кутової швидкості обертання валу, до якого кріпиться певна частина кінцівки робота, у кожен момент часу. Далі лишається знайти добуток цих параметрів — і потужність відома.

3.8 Програмування визначення динамічних процесів та споживаної потужності моделі робота Darth EvadeR

Слідуючи висновку з підрозділу 3.7, у визначенні потужності робота використовуватиметься саме формула (3.2).

Обертаючий момент та кутова швидкість були знайдені з блоку Revolute Joint, котрий можна побачити виділеним червоним на рисунку 3.14. Як вже згадувалося у минулих розділах, він відіграє роль суглобу і дозволяє встановлювати зв'язок між двома фреймами; в даному випадку це колінний суглоб Knee Joint, котрий з'єднує гомілку (Lower Leg) та стегно (Upper Leg). Обертаючий момент у блоці Knee Joint позначено як t , а кутова швидкість як w . Параметр q вказує на кут між стегном та гомілкою, що динамічно змінюється при ходьбі.

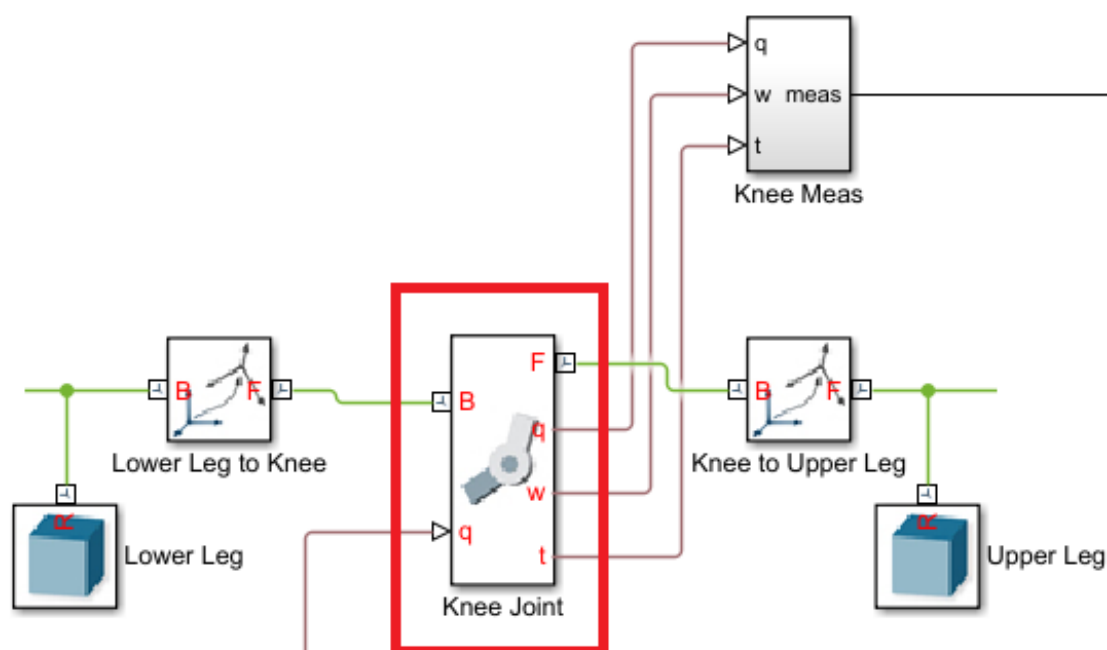


Рисунок 3.14 — Ділянка моделі у Simulink з виділеним блоком Revolute Joint

Значення параметрів t та w кожні 0.1333 с моделювання записувалися до спеціальних змінних, а тому після написання нескладного скрипту було з легкістю знайдено потужність у кожен зафіксований момент часу. Надалі необхідно було з

масиву отриманих даних розрахувати споживану роботом потужність за весь час моделювання. Для виконання цієї задачі були прийняті наступні кроки:

1. Була розділена ходьба на дві фази: фаза 1 — коли робот йде прямолінійно, фаза 2 — коли робот здійснює обхід перешкоди.
2. Було взяте середнє значення потужності для кожного приводу робота за один цикл ходьби, який триває 0.8 с. Взяті значення були просумовані, в результаті чого було отримано середнє значення потужності, що споживає робот за один цикл ходьби у першій фазі. Потім за формулою (3.3) була знайдена сумарна потужність, котра була витрачена роботом на відпрацювання ходьби у першій фазі:

$$P_{\phi} = \frac{t_{\phi} \cdot P_{\text{сум.за цикл}}}{0.8}, \quad (3.3)$$

де t_{ϕ} — загальний час перебування робота у конкретній фазі, с;

$P_{\text{сум. за цикл}}$ — сумарне споживання потужності роботом протягом одного циклу ходьби.

3. Дії з пункту 2 були застосовані і для фази 2. Отриманий результат — сумарна потужність, котра була витрачена роботом на ходьбу у другій фазі.

Сума знайдених потужностей з другого пункту та третього і є тією самою потужністю, котра витрачається роботом за весь час моделювання.

Для демонстрації працездатності програми перевіримо скрипт на прикладі з розділу 3.7 (рисунок 3.15, 3.16).

```

*****
Right Ankle Z power before 0.5s is 0 Watt
Right Ankle Z power from 0.55s to 1.3s is 0 Watt
Energy consumption of Right Ankle Z during ZERO phase is 0 Watt
Right Ankle Z power during side walk cycle: 36.3368 Watt
Energy consumption of Right Ankle Z during ONE phase is 241.1873 Watt
FULL Energy consumption of Right Ankle Z is 241.1873 Watt
Walking cycles (zero): 17.6617
Walking cycles (one): 6.6375
*****
Right Hip Y power before 0.5s is 4.9242e-17 Watt
Right Hip Y power from 0.55s to 1.3s is 8.0477e-19 Watt
Energy consumption of Right Hip Y during ZERO phase is 1.4214e-17 Watt
Right Hip Y power during side walk cycle: 227.0944 Watt
Energy consumption of Right Hip Y during ONE phase is 1507.3497 Watt
FULL Energy consumption of Right Hip Y is 1507.3497 Watt
Walking cycles (zero): 17.6617
Walking cycles (one): 6.6375
*****
Right Ankle power before 0.5s is 60.7833 Watt
Right Ankle power from 0.55s to 1.3s is 44.9028 Watt
Energy consumption of Right Ankle during ZERO phase is 793.0596 Watt
Right Ankle power during side walk cycle: 1.7623e-14 Watt
Energy consumption of Right Ankle during ONE phase is 1.1697e-13 Watt
FULL Energy consumption of Right Ankle is 914.6261 Watt
Walking cycles (zero): 17.6617
Walking cycles (one): 6.6375
*****
Right Knee power before 0.5s is 107.7764 Watt
Right Knee power from 0.55s to 1.3s is 3.5463e-12 Watt
Energy consumption of Right Knee during ZERO phase is 6.2634e-11 Watt
Right Knee power during side walk cycle: 1.8865e-12 Watt
Energy consumption of Right Knee during ONE phase is 1.2522e-11 Watt
FULL Energy consumption of Right Knee is 107.7764 Watt
Walking cycles (zero): 17.6617
Walking cycles (one): 6.6375
*****
Right Hip power before 0.5s is 61.9722 Watt
Right Hip power from 0.55s to 1.3s is 272.6708 Watt
Energy consumption of Right Hip during ZERO phase is 4815.8334 Watt
Right Hip power during side walk cycle: 65.7834 Watt
Energy consumption of Right Hip during ONE phase is 436.6402 Watt
FULL Energy consumption of Right Hip is 5376.418 Watt
*****
Left Hip Y power before 0.5s is 1.5124e-17 Watt
Left Hip Y power from 0.55s to 1.3s is 3.1366e-17 Watt
Energy consumption of Left Hip Y during ZERO phase is 5.5397e-16 Watt
Left Hip Y power during side walk cycle: 99.0123 Watt
Energy consumption of Left Hip Y during ONE phase is 657.1986 Watt
FULL Energy consumption of Left Hip Y is 657.1986 Watt
*****
Left Ankle power before 0.5s is 55.2307 Watt
Left Ankle power from 0.55s to 1.3s is 56.8695 Watt
Energy consumption of Left Ankle during ZERO phase is 1004.4127 Watt
Left Ankle power during side walk cycle: 1.3777e-11 Watt
Energy consumption of Left Ankle during ONE phase is 9.1447e-11 Watt
FULL Energy consumption of Left Ankle is 1114.8741 Watt
*****

```

Рисунок 3.15 – Перший фрагмент відпрацювання скрипту

Також, написаний скрипт дозволяє виводити графіки, які демонструють кутове прискорення та обертальний момент суглобу, споживання потужності у кожен момент часу, стани робота (нуль – коли робот не виявляє перешкоди, одиниці – коли перешкода виявлена і робот у стані її обходження). Усі чотири графіки зображені на рисунку 3.17, розглядається стегновий суглоб лівої ноги.

Графік споживаної потужності (третій) зі встановленим обмеженням для максимально дозволеної потужності у розмірі 500 Вт. Цей крок необхідний для оптимальності виведення величин.

Програмний код, описаний у цьому розділі, можна знайти у Додатку Б.

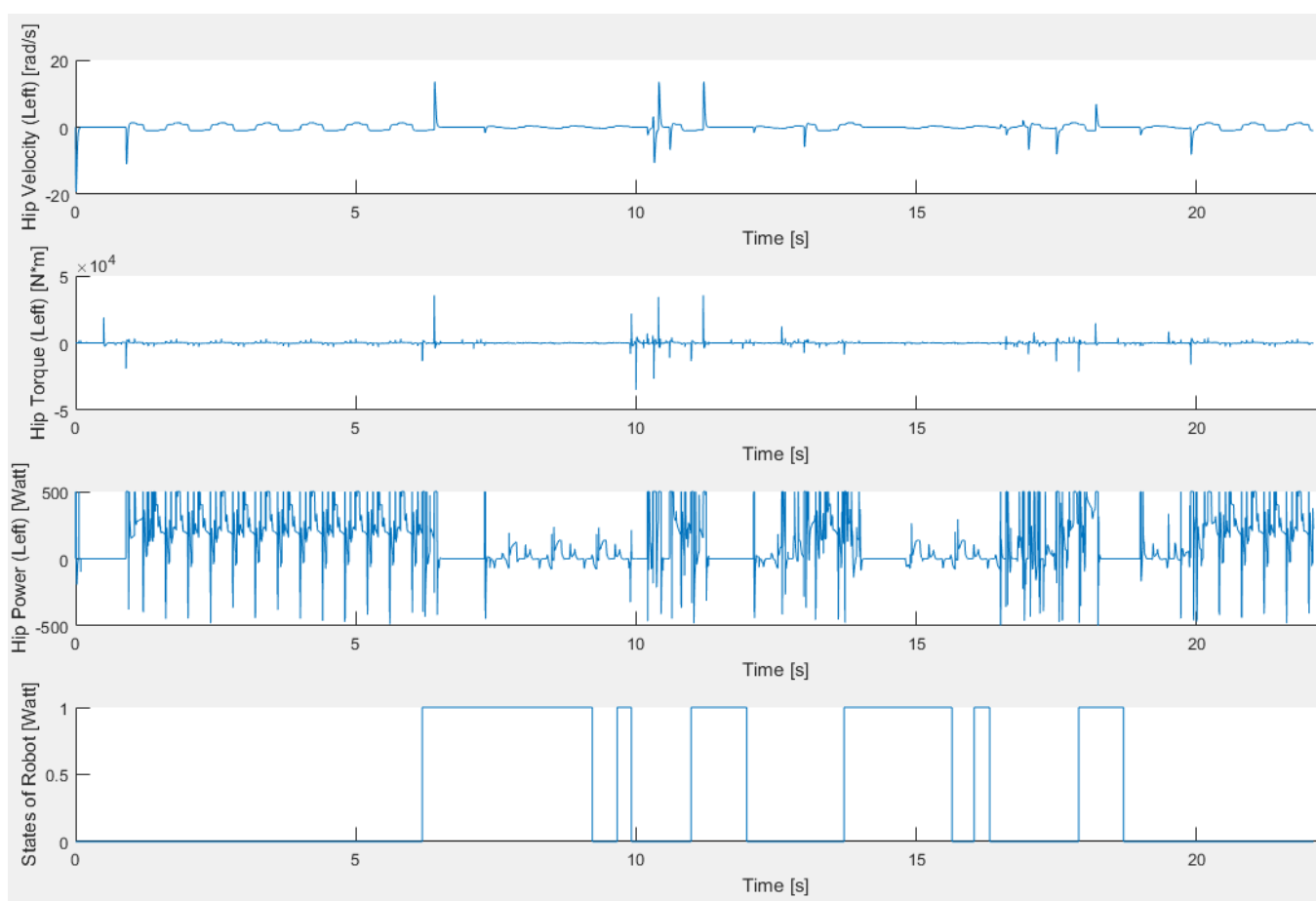


Рисунок 3.17 – Графіки залежності кутової швидкості, обертального моменту суглобу, споживання потужності та станів робота від часу моделювання для стегнового суглобу лівої ноги

Для того, щоб надати повне уявлення того, що в дійсності відбувається з роботом у даному прикладі, наведемо додаткові графіки. На рисунку 3.18 зображено зміну відстані між роботом та перешкодою у відповідності до часу моделювання, а рисунок 3.19 відображає пройдену роботом відстань за весь час моделювання.

Відповідно до рисунку 3.18: початкова відстань між роботом і перешкодою 2.5 м (від центру перешкоди до центру тулуба Darth Evader'a). Після 0.5 секунд групування робот починає ходьбу і через 6.5 с виявляє перед собою заваду на відстані 1 м; він групується і починає її обходити. За 13.5 секунд її обходить і вже на 20-й секундні відновлює свій прямолінійний рух.

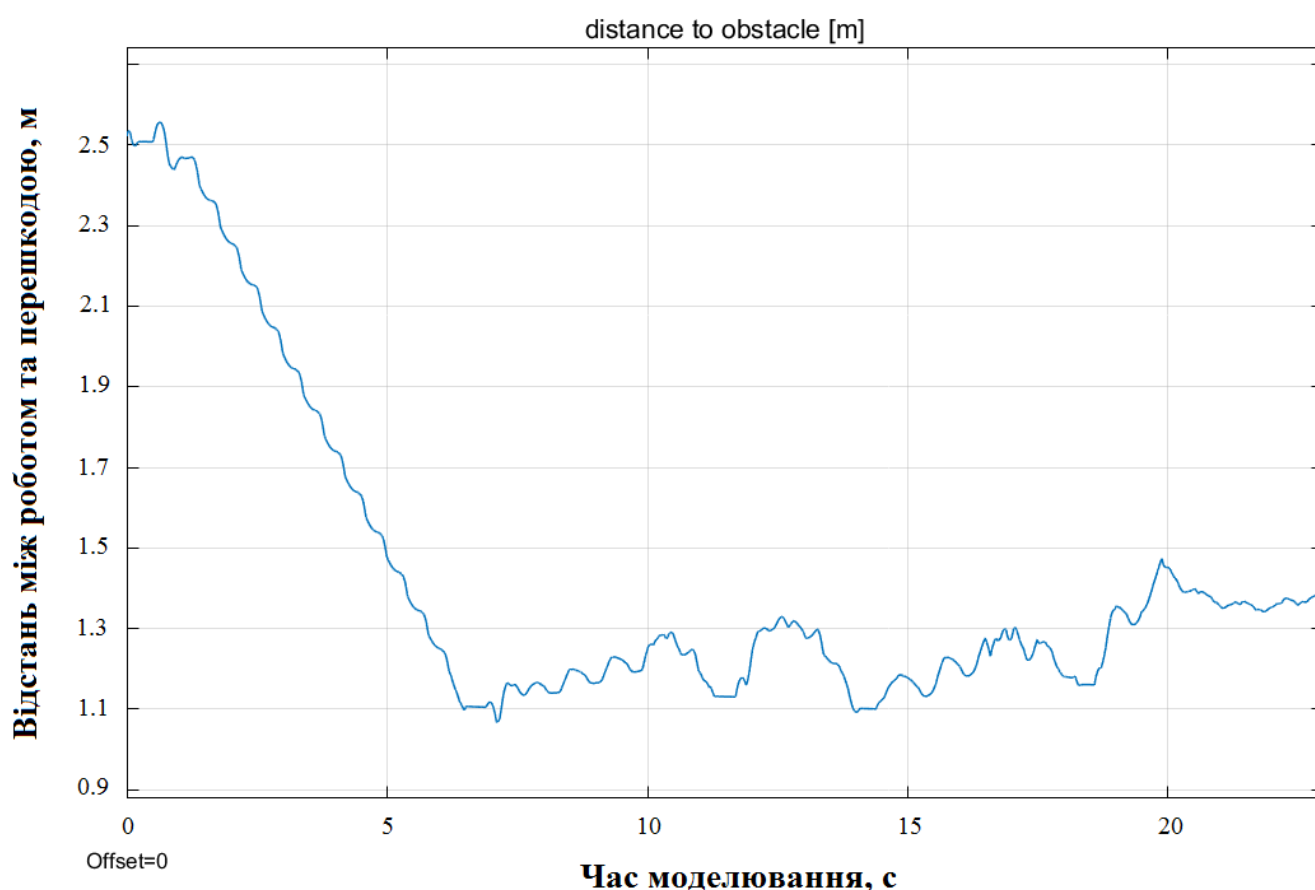


Рисунок 3.18 – Графік відстані між роботом та перешкодою

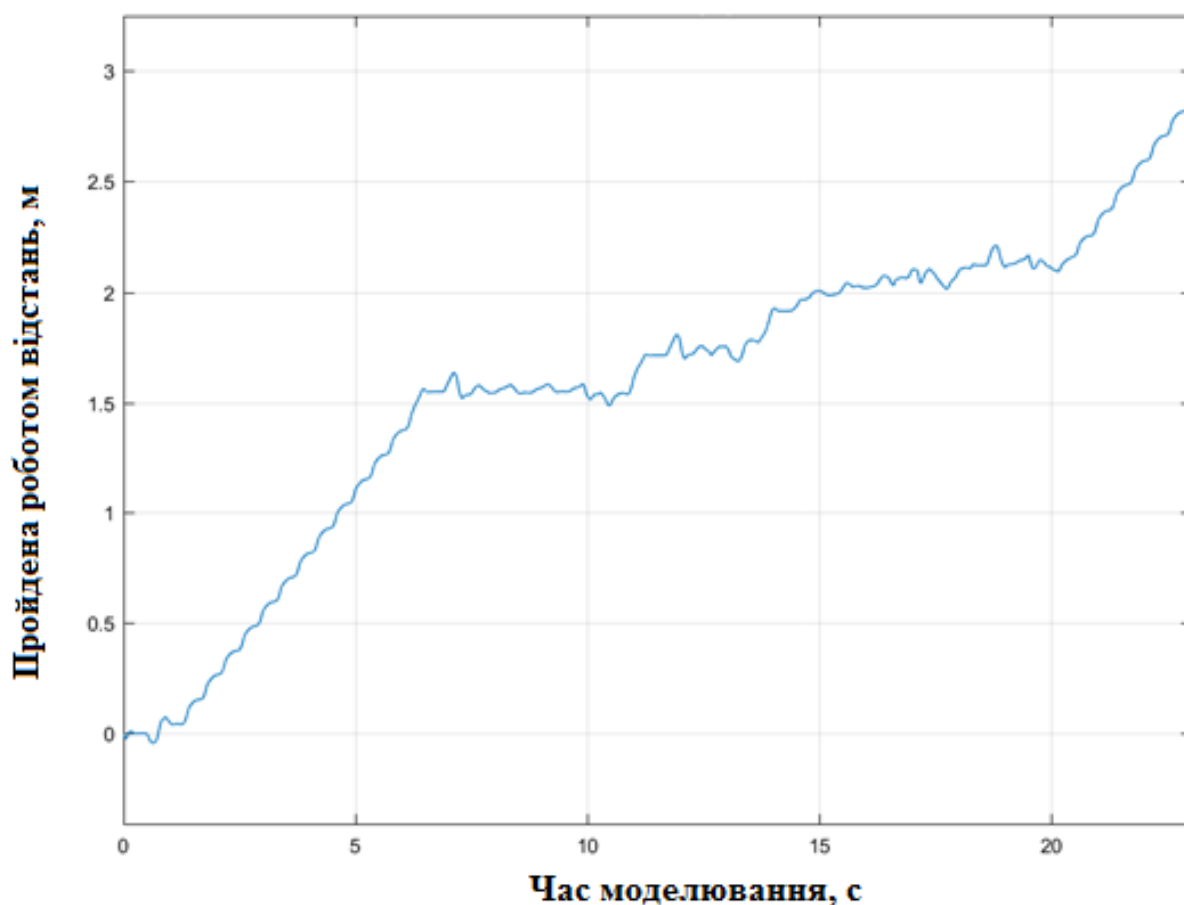


Рисунок 3.19 – Відстань, пройдена роботом за весь час моделювання

Рисунок 3.19 зображує графік, котрий повністю підтверджує висновки, зроблені у минулому абзаці. Загалом за час моделювання робот проходить близько 3-х метрів (з урахуванням того, що перед ним стоїть перешкода).

На рисунку 3.20 зображено робота у вікні Mechanics Explorer відповідно до тих часових проміжків, які були відмічені при розгляді рисунків 3.18 та 3.19.

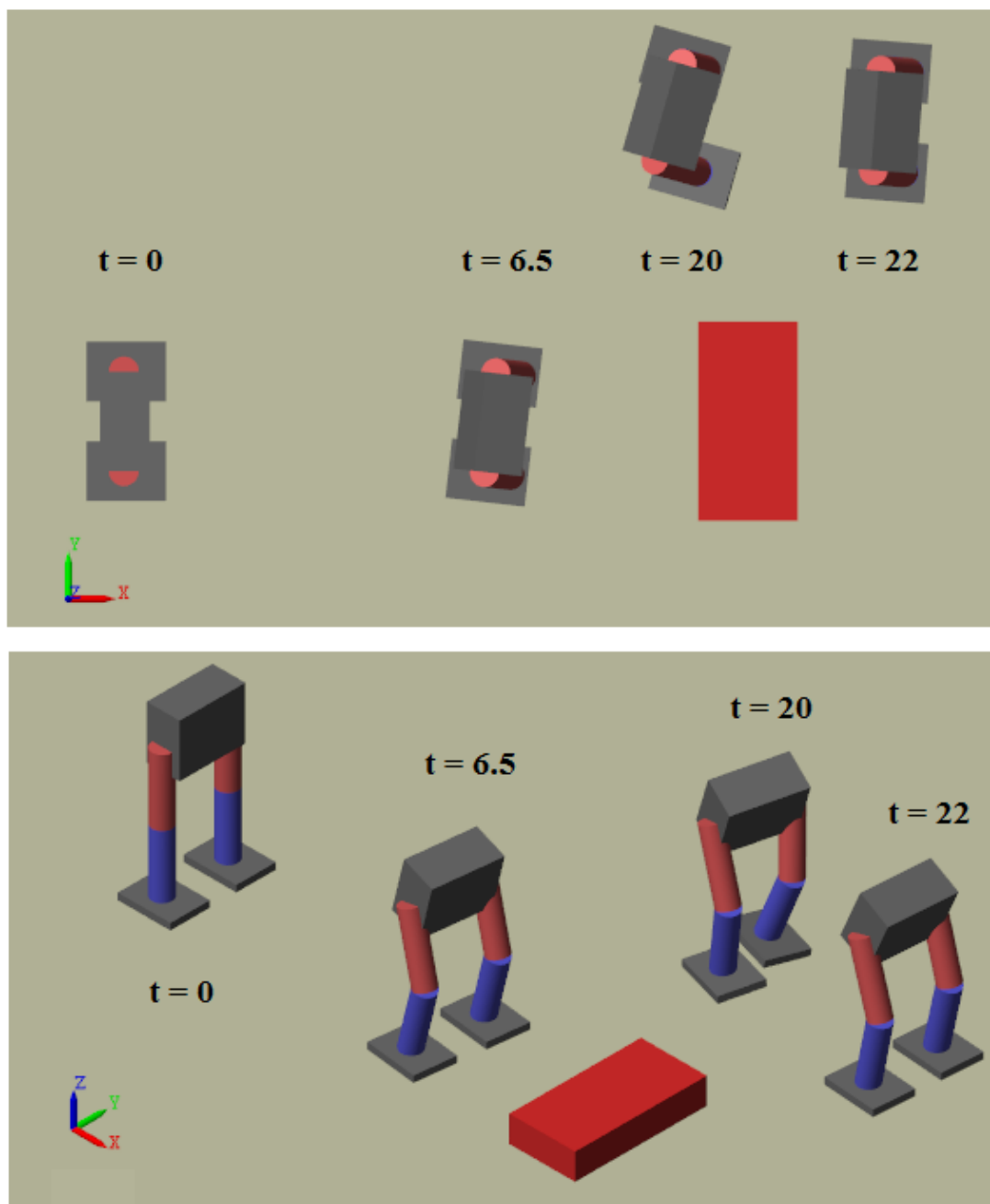


Рисунок 3.20 – Модель робота у Mechanics Explorer у різні моменти часу

Висновки до розділу 3

В даному розділі досягнуто наступні важливі цілі:

- описано усі виконані кроки модернізації, котрі дозволили наблизити авторську модель робота до людських розмірів, а також надати їй

можливість здійснювати обхід перешкоди з подальшим відновленням прямолінійної ходьби; модернізована модель отримала назву Darth EvadeR;

- подано опис загального алгоритму функціонування моделі робота Darth EvadeR;
- розроблено блок-схему алгоритму функціонування робота;
- подано пояснення логіки переміщення робота при виявленні перешкоди, котра розроблена за допомогою інструментів Simulink, а також наведено пояснення щодо призначення кожного блоку, який бере участь у реалізації логіки відпрацювання моделі робота;
- розроблено скрипт для визначення середньої швидкості моделі робота протягом усього часу моделювання;
- визначена формула розрахунку потужності через обертальний момент та кутову швидкість приводу (суглобу) робота Darth EvadeR;
- розроблено скрипт для визначення середнього значення споживаної потужності роботом.

4 АНАЛІЗ ТА ОБРАХУВАННЯ ЕФЕКТИВНОСТІ МОДЕЛІ КРОКУЮЧОГО РОБОТА DARTH EVADER

4.1 Загальні теоретичні відомості про ефективність переміщення живих істот та штучних механізмів

Під час розробки роботів, котрі вміють впливати на своє просторове положення, нерідко з'являється необхідність у визначенні своєрідної ефективності механізму. При цьому необхідно використовувати такий критерій, котрий би давав автору робота не лише чітке розуміння ефективності своєї розробки, але й був би достатньо універсальним для порівняння робота з іншими аналогами, або навіть і з іншими типами роботів, котрі відрізняються кількістю кінцівок та їх типом, кількістю степенів свободи, механізмами активування руху, типом живлення тощо.

На теперішній час, такий критерій було розроблено, його назва — Cost of Transport (перекладається з англ. як «вартість транспортування»; найчастіше вживається абревіатура CoT). Він використовується для визначення ефективності роботів та інших рухомих транспортних засобів (включаючи живих істот), і при цьому є безрозмірним енергетичним критерієм, котрий вказує, скільки енергії необхідно витратити на переміщення одиниці ваги на одиницю відстані [41]. CoT визначається відношенням енерговитрат до добутку маси тіла істоти, гравітаційної сталої та пройденого істотою шляху. Сама формула має наступний вигляд (4.1):

$$CoT = \frac{E}{mgd}, \quad (4.1)$$

де E — кількість енергії, витрачена істотою на подолання відстані d , Дж;

m — маса істоти, кг;

g — гравітаційна стала, 9.81 м/с^2 ;

d — відстань, м.

Також, часто використовується й інша інтерпретація даної формули — вона пов'язана з діаграмою Кармана-Габріелі [42]. Відповідно до інформації за посиланням, CoT ще називається питомою силою тяги, питомим опором або індексом енергії і може обчислюватися відповідно до наступної формули (4.2):

$$CoT = \frac{P}{mgv}, \quad (4.2)$$

де P — потужність робототехнічної системи, Вт;

v — швидкість системи, м/с.

Таким чином, формула (4.2) опирається на потужність P , якою треба забезпечити робота, аби він рухався із швидкістю v . Саме тому (4.2) є зручнішою за (4.1) для обчислень, пов'язаних з роботами та іншими штучними механізмами.

У розділі 3 було вирішено використовувати формулу 3.2 для знаходження середньої потужності кожного окремого приводу робототехнічної системи, після чого отримані значення сумуються і підставляються до формули (4.2).

Щоб отримати більш зрозуміліше уявлення щодо показника CoT, доцільно розглянути CoT існуючих роботів. Оскільки CoT є безрозмірним параметром, його можна використовувати у дуже широкому спектрі досліджень, до яких входять не лише ходячі роботи, але й люди, наземні та водні тварини, птахи, а також інші транспортні системи.

Вже на момент 1975-го року мічиганським доктором зоології В. А. Такером було наведено перший графік CoT, який відображав даний показник для різного роду живих істот та створених людиною механізмів [43]. Графік зображено на рисунку 4.1. На ньому автор за допомогою логарифмічної шкали спробував розмістити різних тварин (риби, птахи, комахи, рептилії, миші, пацюки, собаки, вівці, качки і т. д.), а також тогочасні досягнення людства (автомобілі Volkswagen та Cadillac, мотоцикл, аероплан, трактор, велосипед тощо). Вісь абсцис позначає масу тіла (у кг), а вісь ординат — показник CoT, розрахований за формулою (4.2).

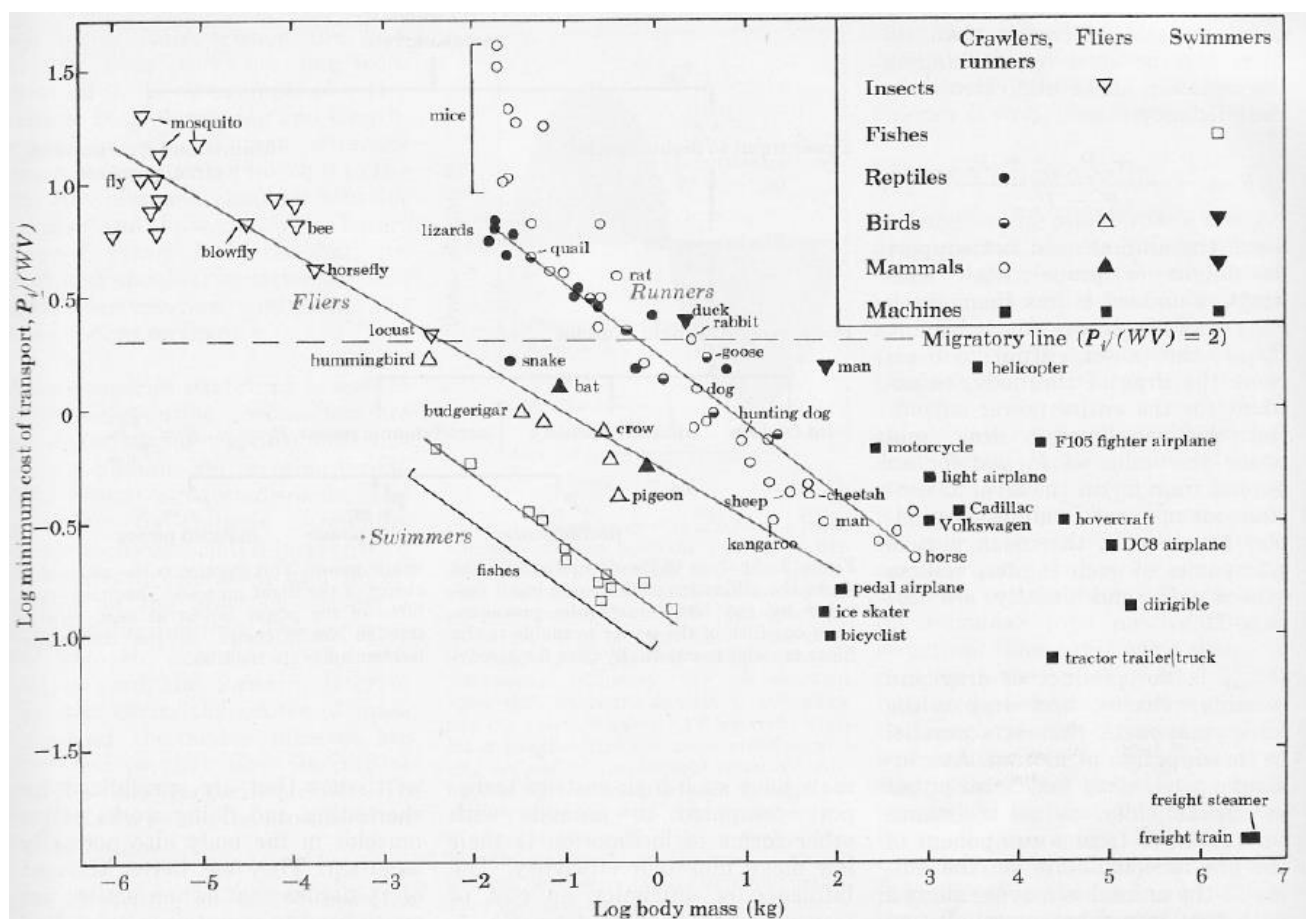


Рисунок 4.1 – Графік, що співставляє мінімальний СоТ різних живих істот та транспортних засобів [43]

На даний момент графік з рисунку 4.1 хоч і є цікавим, однак він не відображає досягнень робототехнічної галузі, а тому несе дуже мало практичної корисності, котру можна було б використати у даній магістерській дисертації.

Насправді, Тао Янг – кандидат у доктори філософії університету Огайо, котрий успішно захистив свою докторську дисертацію у 2007-му році – презентував у своїй роботі версію графіку СоТ, котра окрім тварин містила ще й показники ефективності відомих на той момент роботів та актуальних транспортних засобів, а також авторського робота ERNIE [41]. Ознайомитися з модернізованим графіком СоТ Тао Янга можна на рисунку 4.2.

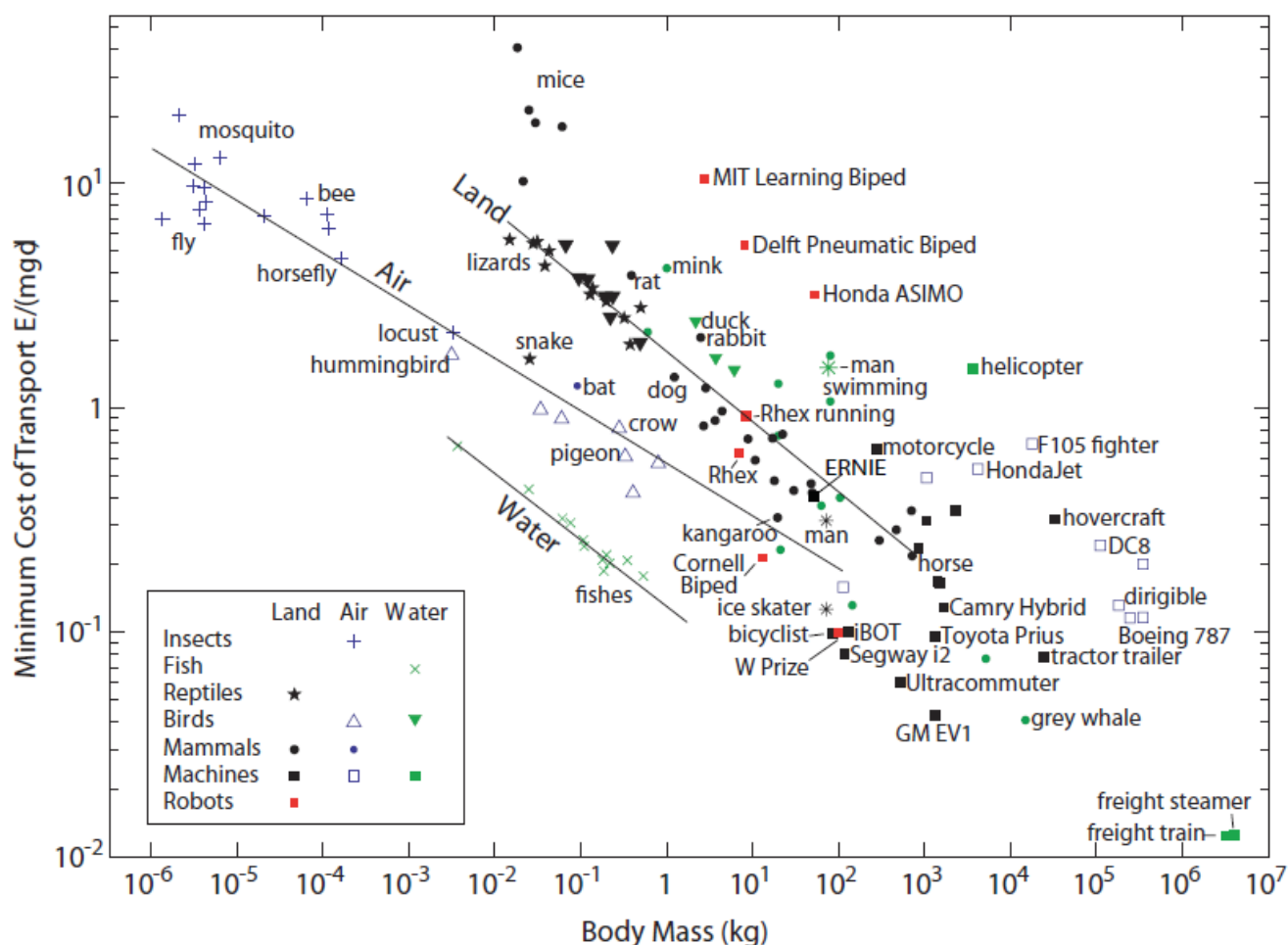


Рисунок 4.2 – Графік, що співставляє мінімальний CoT різних живих істот, транспортних засобів та роботів [41]

Відповідно до рисунку 4.2, роботи мають значення CoT приблизно від 0.1 до 12. Однак, це не є певним еталонним діапазоном. Відповідно до ресурсу [44], чотириногий робот Gregor I вагою 1 кг має показник ефективності рівний 70, а мікроробот HAMR2 (0.002 кг) – 128.

У кожного механізму своє значення CoT, але чим він нижчий, тим краще, оскільки низьке значення свідчить про більш раціональне використання енергії на переміщення на задану відстань.

4.2 Обчислення CoT робота та знаходження оптимального значення

Усі процедури обчислення були пояснені, тому визначимо показник CoT, а також виконаємо декілька експериментів, аби знайти більш оптимальне значення індексу ефективності.

Важливо розуміти, що виміри швидкості та енерговитрат виконуються за наступних умов: робот має пройти 1.5 м, зупинитися перед перешкодою на відстані 1 м від неї (загальна довжина перешкоди — 0.8 м), обійти приблизно 0.4 м перешкоди, відновити прямолінійний рух і зайти за перешкоду так, щоб праворуч та ліворуч від робота був вільний простір (зазвичай це 3-4 цикли ходьби після уникнення перешкоди або ж 0.46 м). Більша наглядність дій вже була продемонстрована на рисунку 3.12 розділу 3.

Отже, Darth EvadeR має наступні параметри:

- вага — 172 кг;
- швидкість — 0.1391 м/с;
- споживана потужність — 13699.8166 Вт.

За основу візьмемо ті параметри робота, котрі розглядалися у розділі 7. Використовуючи формулу (4.2) отримуємо (4.3):

$$CoT = \frac{13699.8166}{172 * 9.8 * 0.1391} = 58.430886 \quad (4.3)$$

Час виконання роботом його задач склав 22.11 с. Змінюємо тривалість циклу ходьби моделі робота, кути згинання-розгинання кінцівок та робимо аналогічні вимірювання сумарної потужності та швидкості ходьби і визначаємо CoT. Усього буде проведено 10 дослідів, серед яких буде визначено найкращі показники якості моделі Darth EvadeR. Насправді, дослідів було значно більше, але не всі з них увінчалися успіхом: в більшості випадків нові значення кутів, що подавалися на приводи робота, призводили до виведення робота з рівноваги на певних моментах ходьби, або ж порушувалася нормальна прямолінійна ходьба Darth EvadeR`а.

Усі результати проведених експериментів знайшли своє відображення у таблиці 4.1.

Таблиця 4.1 — Результати досліджень CoT моделі робота Darth EvadeR

№ досліджу	P, Вт	v, м/с	CoT	Що змінено
1	14010.2689	0.093	89.373	Збільшено кути відхилення стегон (фаза 1), приведено праву стопу за віссю Y до кута -4° (фаза 2)
2	20740.6046	0.11707	105.1044905	Збільшено кути відхилення стегон, колін та стоп (фаза 1), приведено праву стопу за віссю Y до кута -14° (фаза 2)
3	13338.8179	0.1299	60.90787	Збільшено цикл ходьби у фазі 2 до 0.95 с
4	14259.2096	0.12914	65.5068777	Збільшено цикл ходьби у фазі 2 до 0.86 с
5	18813.5754	0.092689	120.417	Збільшено цикл ходьби у фазі 1 до 0.83 с, а у фазі 2 до 0.86 с
6	19082.3566	0.187889	60.2526	Зменшено цикл ходьби у фазі 1 до 0.52 с
7	18016.4663	0.253515	42.16	Зменшено цикл ходьби у фазі 1 до 0.47 с, а у фазі 2 збільшено до 0.81 с
8	19102.2827	0.204609	55.38677	Зменшено цикл ходьби у фазі 1 до 0.47 с, а у фазі 2 до 0.77 с
9	11090.1607	0.14322	45.94	Збільшено цикл ходьби у фазі 1 до 0.83 с, а у фазі 2 до 0.81 с
10	14957.636	0.10653	83.298	Збільшено цикл ходьби у фазі 1 до 0.86 с, а у фазі 2 до 0.81 с

Як можна бачити з таблиці 4.1, оптимальне значення CoT вийшло у сьомому досліді, де було зменшено цикл ходьби у фазі 1 з 0.8 с до 0.47 с, а у фазі 2 — збільшено з 0.8 с до 0.81. Фаза 1 позначає прямолінійну ходьбу робота, фаза 2 — стан, коли робот займається обходженням перешкоди (це розглядалося у розділі 3, підрозділ 3.8). Дійсно, енергоспоживання виросло на 4.3 кВт, проте швидкість суттєво зросла.

Таблиця 4.2 показує час виконання моделлю робота своїх задач у кожному з дослідів.

Таблиця 4.2 — Час виконання моделлю робота своїх задач відповідно до дослідів

№ дослідів	1	2	3	4	5	6	7	8	9	10
Час, с	21.89	24.59	27.38	24.61	31.74	15.24	14.01	17.09	17.64	29.75

Відповідно до таблиці 4.2, найкращий час виконання роботом покладених на нього обов'язків також належить роботі з параметрами із дослідів під номером 7.

Визначимо, який виграш дає знайдений оптимальний показник CoT за часом виконання завдання, швидкістю та споживаною потужністю у порівнянні з оригінальними показниками, що задіяні у формулі (4.3). Таким чином, за формулою (4.4) знаходимо зміну часу:

$$t^* = \left| 100\% - \frac{14.01 \text{ с} * 100\%}{22.11 \text{ с}} \right| = |100\% - 63.36\%| = 36.64\%. \quad (4.4)$$

Отже, час виконання роботом завдання покращився на 36.64%. За формулою (4.5) обчислюємо зміну швидкості:

$$v^* = \left| 100\% - \frac{0.2535 \frac{\text{м}}{\text{с}} * 100\%}{0.1391 \frac{\text{м}}{\text{с}}} \right| = |100\% - 182.25\%| = 82.25\%. \quad (4.5)$$

[illegible]

Рисунок 4.4 – Другий фрагмент результату відпрацювання скрипту для знаходження потужності

Також, слід навести графік швидкості моделі робота (рисунок 4.5) з оптимальними параметрами.

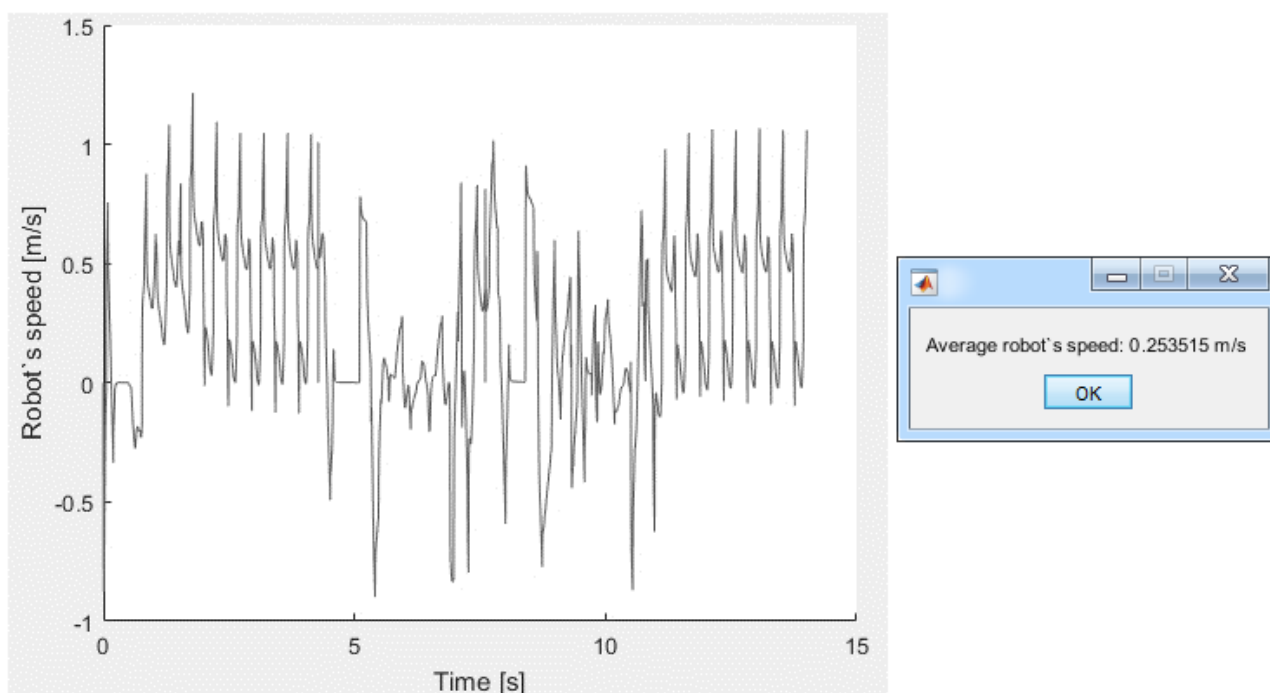


Рисунок 4.5 – Швидкість виконання моделлю робота завдання та відповідний графік швидкості у кожен момент часу

Як можна бачити з рисунку 4.5, перші 0.5 секунд робот знаходиться у стані групування (режим Grouping), після чого починає йти (режим Walking Forward). Приблизно на 4-й секунді він натикається на перешкоду, врівноважується перші 0.5 секунд (Grouping), і до 11-ї секунди її обходить (режим Side Walking), що помітно по великим нестійким коливанням. Після 11-ї секунди робот відновлює прямолінійну рівномірну ходьбу, а саме моделювання завершується після того, як Darth EvadeR заходить за перешкоду.

На рисунку 4.6 зображено зміну відстані між роботом та перешкодою за весь час моделювання, а на рисунку 4.7 – пройдений роботом шлях відповідно до часу моделювання.

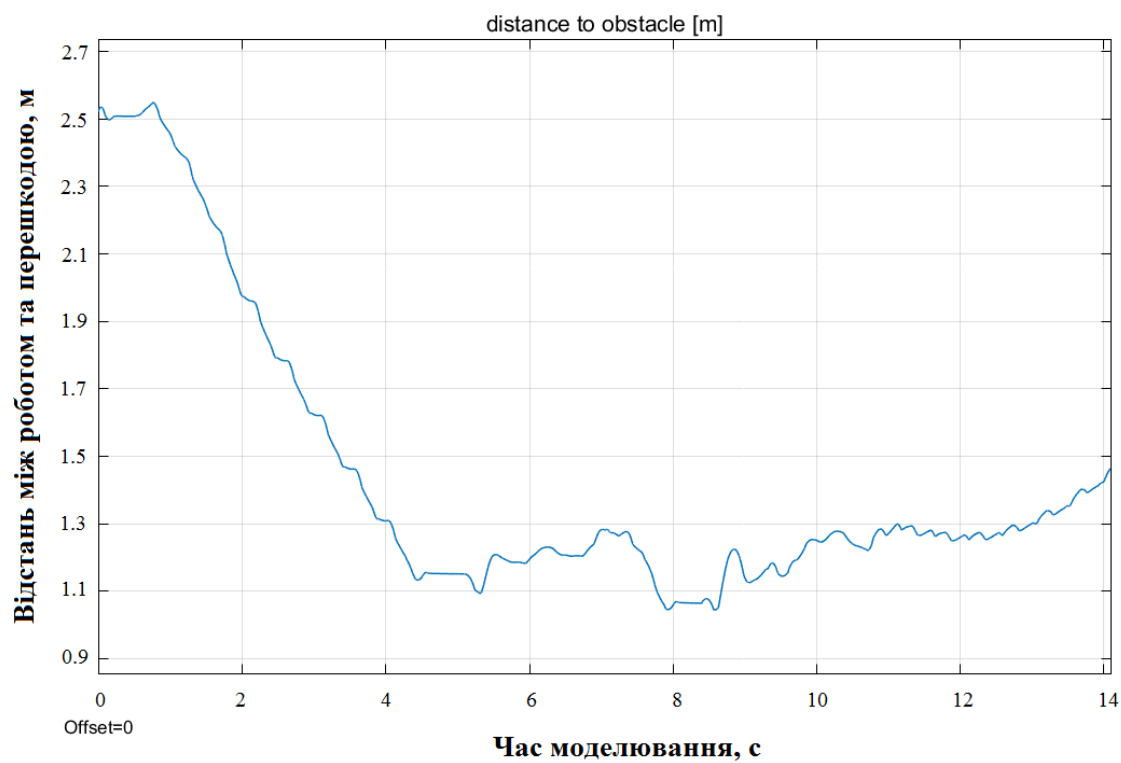


Рисунок 4.6 – Графік відстані між роботом та перешкодою

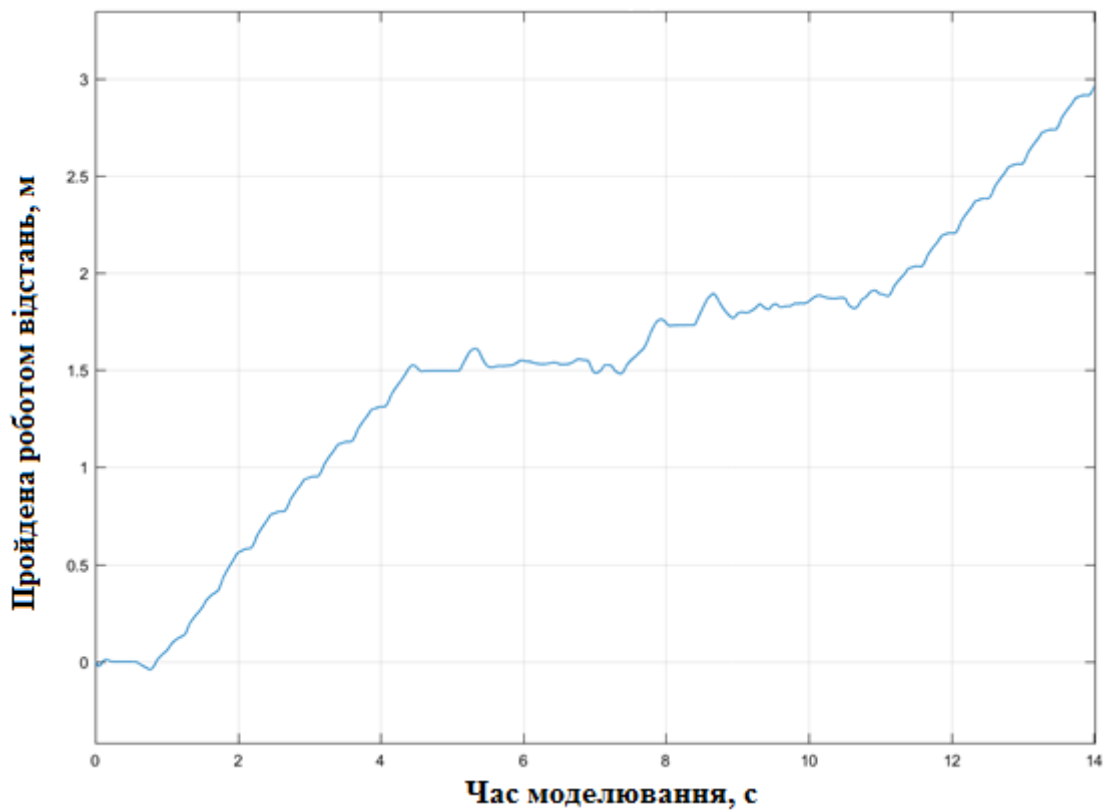


Рисунок 4.7 – Графік відстані, пройденної роботом за весь час моделювання

З рисунку 4.6 видно: Darth EvadeR розпочинає свій рух на відстані 2.5 м від перешкоди. Перші 0.5 секунд він стоїть, а саме – перебуває у стані Grouping. Після цього робот розпочинає рух, проходить 1.5 м і при відстані між роботом та перешкодою у розмірі 1 м зупиняється (на ходьбу було витрачено 3.5 с). Розпочинається друга фаза – робот групується і починає обходити перешкоду. На 11-й секунді обхід закінчується, робот фіксує вільний простір перед собою і продовжує рухатися прямолінійно рівномірно, збільшуючи дистанцію між завадою та собою.

Графік на рисунку 4.7 також підтверджує усі вище наведені тези: робот проходить 1.5 м, обходить заваду, а далі відновлює і успішно продовжує прямолінійний рух вперед.

Підводячи підсумки, можна сказати, що оптимальне значення CoT супроводжується кращою швидкістю, кращим часом виконання роботом завдання та відчутним програвом у споживанні енергії.

4.3 Порівняння отриманого оптимального CoT з існуючими роботами

Для того, щоб зобразити CoT моделі Darth EvadeR`а та порівняти цей показник з показником ефективності існуючих роботів, візьмемо за основу графік, що був наведений у джерелі [45]. Дане джерело датується 2017-м роком та містить відомості щодо найбільш сучасних та відомих роботів з кінцівками, а отже, є актуальним та доцільним у використанні в даному розділі дисертації.

Якщо розширити графік з джерела та додати робота Darth EvadeR, буде отриманий графік, зображений на рисунку 4.8. Логарифмічна шкала в даному випадку використовується для того, щоб було зручно розмістити на одній площині дані, котрі можуть відрізнятися декількома порядками. Наприклад, CoT людини стандартної ваги (80 кг) дорівнює 0.2, в той час як вже згаданий мікроробот HAMR2 вагою 0.002 кг має показник ефективності рівний 128.

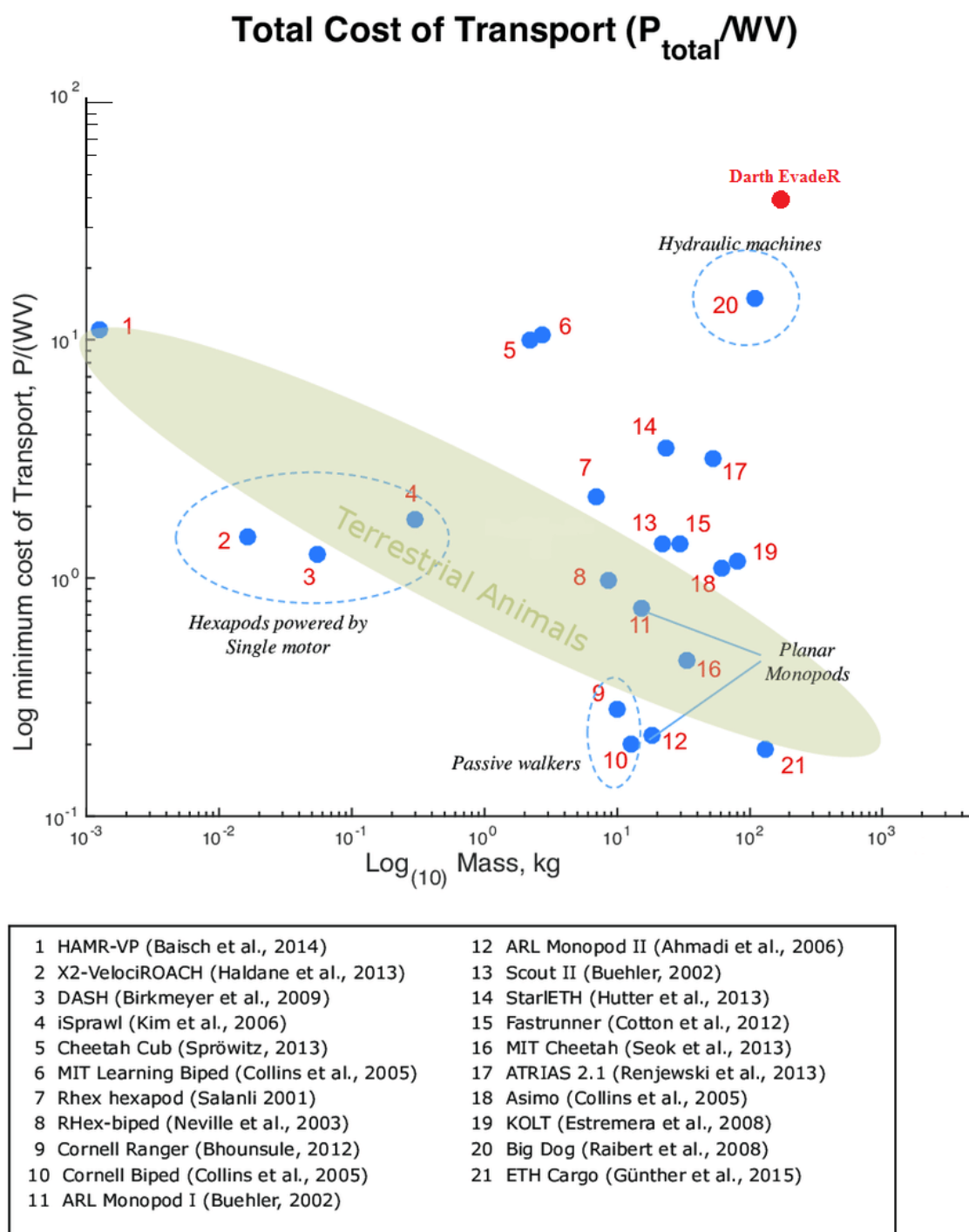


Рисунок 4.8 - Показник CoT (P/mgv) роботів з кінцівками у порівнянні з CoT тварин, а також з показником CoT моделі робота Darth EvadeR [45]

Більш детальне порівняння параметру CoT моделі Darth EvadeR з іншими відомими роботами з кінцівками краще зобразити у табличному вигляді. Власне, таблиця 4.3 демонструє показник ефективності наступних робототехнічних

екземплярів: двоногий різновид робота RHex [46], двоногий DURUS [16-18], шестиногий мікроробот DASH, двоногий ATRIAS, шестиногий мікроробот HAMR2, двоногий Asimo, двоногий Atlas, чотириногий Big Dog та, відповідно, Darth EvadeR. Дані взяті з джерела [44].

Таблиця 4.3 — CoT різних відомих роботів, а також CoT моделі Darth EvadeR

Назва	RHex	DURUS	DASH	ATRIAS	HAMR2
Маса, кг	9	79.5	0.0062	62	0.002
CoT	1	1.61	147	1.13	128

Продовження таблиці 4.3

Назва	ASIMO	Atlas	Big Dog	Darth EvadeR
Маса, кг	50	102	110	172
CoT	3.2	5	15	42.16

Як можна бачити з рисунку 4.8 та таблиці 4.3, робот Darth EvadeR дійсно має дещо високий показник CoT, однак, це ціна можливості обходити перешкоду, будучи двоногим крокуючим роботом, якою володіють в основному високовартісні мультифункціональні гуманоїдні роботи як-от японський ASIMO, американський Atlas та інші. При цьому подолання перешкод є все ж прерогативою роботів з більшою кількістю нижніх кінцівок, оскільки їм значно легше втримувати баланс та уникати перешкод, але вони при цьому не можуть бути повноцінними помічниками людини через особливості своєї побудови.

З огляду на таблицю 4.3 можна сказати, що подібний показник ефективності моделі Darth EvadeR не є незадовільним, оскільки велика вага робота є катализатором високих затрат на потужність, враховуючи ще й те, що модель ДКР витрачає порівняно мало часу на зближення із завадою та її обходження. Отриманий оптимальний CoT свідчить про те, що модель робота Darth EvadeR не є ідеальною і

потребує подальшого вдосконалення як-от зменшення ваги робота, розробка більш досконалого алгоритму пересування тощо.

Висновки до розділу 4

В даному розділі було обчислено безрозмірний показник ефективності моделі двоногого крокуючого робота Darth EvadeR, який позначається як CoT (англ. «Cost of Transport»). Більше того, була проведена серія експериментів, котра допомогла виявити оптимальне значення CoT. Воно характеризувалося збільшенням швидкості переміщення робот на 82.25%, зменшенням часу на виконання завдання на 36.64%, однак дало програв у споживанні потужності, котра зросла на 31.5%.

Оптимальне значення CoT було порівняно з існуючими роботами з кінцівками, в результаті чого був зроблений висновок: модель потребує подальшого вдосконалення. Цього можна досягти шляхом реалізації наступних двох пунктів: зменшення ваги робота та вдосконалення алгоритму переміщення в обох фазах.

ВИСНОВКИ

В результаті виконання магістерської дисертації вдосконалено функціональні можливості моделі двоногого крокуючого робота, створеної за допомогою пакету прикладних програм MATLAB та інтерактивного середовища Simulink, а саме:

1. Розроблено вдосконалену схему моделі ДКР на базі авторської моделі Себастьяна Кастро.
2. Збільшено габарити робота, а також його масу, аби ці показники більше відповідали людським.
3. Розширено динаміку гомілково-стопних та стегнових суглобів робота за рахунок збільшення кількості степенів свободи моделі робота до 12-ти.
4. Розроблено динаміку обходження двоногим роботом перешкоди, виявленої на його шляху, з подальшим відновленням початкової прямолінійної ходьби.

Технічні параметри моделі Darth EvadeR:

- вага – 172 кг;
- зріст – 100 см;
- швидкість руху – 0.1381 м/с;
- енергоспоживання – 13.699 кВт (за час виконання роботом завдання з наближення до завади та її обходження);
- кількість степенів свободи – 12.

Окрім змін у функціональній частині моделі, виконано наступне:

1. Розроблено структурну схему модернізованої моделі робота, яка дозволяє зрозуміти склад моделі, взаємозв'язки та призначення основних її частин.
2. Розроблено алгоритм функціонування моделі робота, який відображає його поведінку.
3. Написано програмний код, котрий дозволяє відслідковувати динамічні процеси моделі робота та її конкретних суглобів, а саме: швидкість переміщення робота у кожен момент часу та кутову швидкість,

обертальний момент, фазу робота і споживану потужність кожним суглобом у кожен момент часу.

4. Досліджено і розраховано ефективність робота – індекс CoT. Обчислений індекс ефективності становить 58.43.
5. Знайдено оптимальний показник CoT, рівний 42.16. При цьому вдалося досягти збільшення швидкості виконання роботою завдання з наближення до перешкоди і її обходження на 82.25%; час виконання завдання при цьому зменшено на 36.64%.

Оптимальне значення ефективності Darth EvadeR`а (CoT) було порівняно зі значеннями інших сучасних роботів, з чого зроблено висновок про необхідність подальшого вдосконалення моделі робота, котре має бути спрямоване на покращення алгоритму пересування робота та зменшення його ваги.

Головна особливість моделі робота Darth EvadeR – вміння обходити перешкоду, яка знаходиться на шляху робота. Загалом, робот вміє перебувати у трьох станах: прямолінійна ходьба, групування, рух ліворуч. Логіка переміщень Darth EvadeR`а реалізована за допомогою стандартних бібліотек середовища Simulink. Також, робот відрізняється тим, що реалізована логіка обходження перешкод не потребує складних алгоритмів або використання додаткових інструментів для самоврівноваження під час ходьби або обходження завади.

За результатами проведених досліджень було опубліковано статтю та тези:

1. Репнікова Н. Б. Вдосконалення функціональних можливостей моделі крокуючого робота / Н. Б. Репнікова, В. А. Сверчков // Проблеми інформатизації та управління: збірник наукових праць. – 2021. — Т. 1, №65 — С. 62-67. <https://doi.org/10.18372/2073-4751.65.15372>.
2. Репнікова Н. Б., Сверчков В. А. Дослідження ефективності двоногого крокуючого робота. Summer InfoCom Advanced Solutions 2021: тези доп. XI Міжнар. наук.-практ. конф., м. Київ, 13-14 трав. 2021 р.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Двуногие ходящие роботы. [Електронний ресурс] : Оглядовий сайт робототехніки — Назва з екрана. Доступ: <http://robotrends.ru/robopedia/dvunogie-hodyashie-roboty>
2. Обзор: шагающие роботы | Triol. [Електронний ресурс] : Офіційний сайт корпорації Triol – Назва з екрана. Доступ: <https://triolcorp.ru/blog/post/obzor-shagayushchie-roboty>
3. Wettergreen D. Robotic walking in natural terrain. Gait planning and behavior-based control for statically-stable walking robots : дис. докт. філос. наук / D. Wettergreen — Pittsburgh PA, 1995. — 134 p.
4. Westervelt E. R. Feedback Control of Dynamic Bipedal Robot Locomotion / E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi та B. Morris. – Boca Raton, FL: CRC Press, 2007. – P. 6-9.
5. Vukobratović M. (2006, September). Humanoid Robotics – Past, Present State, Future. In 2006 4th Serbian-Hungarian Joint Symposium on Intelligent Systems (SISY 2006) (pp. 13-31). SISY.
6. Humanoid History – Biped Walking Robot. [Електронний ресурс] : Офіційний сайт університету Васеда – Назва з екрана. Доступ: http://www.humanoid.waseda.ac.jp/booklet/kato_4.html
7. Vukobratović M. and Juricić D. (1969, January). Contribution to the synthesis of biped gait, In Proc. IFAC Symposium on Technical and Biological Problem on Control, IEEE Transactions on Biomedical Engineering (pp. 1-6), 1968. <https://doi.org/10.1109/TBME.1969.4502596>
8. Humanoid History – WABOT. [Електронний ресурс] : Офіційний сайт університету Васеда – Назва з екрану. Доступ: http://www.humanoid.waseda.ac.jp/booklet/kato_2.html
9. Wabot 2 – ROBOTS: Your Guide to the World of Robotics. [Електронний ресурс] : Світлина з сайту міжнародної організації IEEE – Назва з екрану. Доступ: <https://robots.ieee.org/robots/wabot/?gallery=photo3>

10. Wabot 2 – ROBOTS: Your Guide to the World of Robotics. [Електронний ресурс] : Світлина з сайту міжнародної організації IEEE – Назва з екрану. Доступ: <https://robots.ieee.org/robots/wabot/?gallery=photo2>
11. Yang X. State of the Art: Bipedal Robots for Lower Limb Rehabilitation / X. Yang, H. She, H. Lu, T. Fukuda and Y. Shen // Appl. Sci. – 2017. – 7(11), 1182. – pp. 8-26. <https://doi.org/10.3390/app7111182>
12. Westervelt E. R. Hybrid Zero Dynamics of Planar Biped Walkers / E. R. Westervelt, J. W. Grizzle, D. E. Koditschek // IEEE Transactions on Automatic Control, 48(1), 2003. – pp. 42-56. <https://doi.org/10.1109/TAC.2002.806653>
13. Honda Global | November 12, 2001 “Honda Introduces New ASIMO Humanoid Robot for Rental Business”. [Електронний ресурс] : Новини офіційного сайту компанії Honda – Назва з екрану. Доступ: <https://global.honda/newsroom/news/2001/c011112-eng.html>
14. Honda Global | November 20, 2000 “Honda Debuts New Humanoid Robot “ASIMO”. [Електронний ресурс] : Новини офіційного сайту компанії Honda – Назва з екрану. Доступ: <https://global.honda/newsroom/news/2000/c001120b-eng.html>
15. Technical Information – Asimo – Honda. [Електронний ресурс] : технічна документація гуманоїдного робота ASIMO – Назва з екрану. Доступ: <https://asimo.honda.com/downloads/pdf/asimo-technical-information.pdf>
16. Durus. [Електронний ресурс] : Оглядовий сайт інженерних новин – Назва з екрану. Доступ: <https://www.wevolver.com/wevolver.staff/durus>
17. Reher J. Realizing Dynamic and Efficient Bipedal Locomotion on the Humanoid Robot DURUS / J. Reher, E. Cousineau, A.E. Hereid, C. M. Hubicki, A. D. Ames // 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016. – pp. 1794-1801. <https://doi.org/10.1109/ICRA.2016.7487325>
18. DURUS: SRI's Ultra-Efficient Walking Humanoid Robot – IEEE Spectrum. [Електронний ресурс] : Інтернет-блог організації IEEE – Назва з екрану. Доступ: <https://spectrum.ieee.org/autoton/robotics/humanoids/durus-sri-ultra-efficient-humanoid-robot>

19. Робот Atlas научился кувыркаться и вставать на руки. [Электронный ресурс] : Интернет-видання N+1 – Назва з екрану. Доступ: <https://nplus1.ru/news/2019/09/24/atlas>
20. Как Boston Dynamics создавала робота Atlas // Как устроен самый крутой робот в мире? [Электронный ресурс] : Відеохостинг YouTube – Назва з екрану. Доступ: <https://www.youtube.com/watch?v=kegGlrYEmM8>
21. How Boston Dynamics Is Redefining Robot Agility – IEEE Spectrum. [Электронный ресурс] : Інтернет-блог організації IEEE – Назва з екрану. Доступ: <https://spectrum.ieee.org/robotics/humanoids/how-boston-dynamics-is-redefining-robot-agility>
22. Atlas Robot. [Электронный ресурс] : Оглядовий сайт інженерних новин – Назва з екрану. Доступ: <https://www.wevolver.com/wevolver.staff/atlas.robot>
23. Modest Debut of Atlas May Foreshadow Age of “Robo Sapiens” – The New York Times. [Электронный ресурс] : Щоденна інтернет-газета – Назва з екрану. Доступ: <https://www.nytimes.com/2013/07/12/science/modest-debut-of-atlas-may-foreshadow-age-of-robo-sapiens.html>
24. Робот-пожарный SAFFiR готовится впервые шагнуть в огонь – DailyTechInfo. [Электронный ресурс] : Інтернет-ресурс про новини науки, технологій та техніки – Назва з екрану. Доступ: <https://dailytechinfo.org/robots/5769-robot-pozharnyy-saffir-gotovitsya-vpervye-shagnut-v-ogon.html>
25. Военные США испытали человекоподобного робота – Naked Science. [Электронный доступ] : Мережеве видавництво – Назва з екрану. Доступ: <https://naked-science.ru/article/tech/voennye-ssha-ispytali-chelovek>
26. SAFFIR The Firefighting Robot. [Электронный ресурс] : Блог компанії Robotiq – Назва з екрану. Доступ: <https://blog.robotiq.com/saffir-the-firefighting-robot>
27. DLR – Institute of Robotics and Mechatronics – Toro. [Электронный ресурс] : Сайт інституту робототехніки та мехатроніки – Назва з екрану. Доступ: <https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11678/#gallery/28597>

28. Engelsberger J. Overview of the torque-controlled humanoid robot TORO / J. Engelsberger, A. Werner, C. Ott, та ін. // 2014 IEEE-RAS International Conference on Humanoid Robots, 2014. – pp. 916-923.
<https://doi.org/10.1109/HUMANOIDS.2014.7041473>
29. Thirteen Advanced Humanoid Robots for Sale Today – Smashing Robotics. [Електронний ресурс] : Інтернет-блог про робототехніку – Назва з екрану. Доступ: <https://www.smashingrobotics.com/thirteen-advanced-humanoid-robots-for-sale-today/>
30. Human-like androids have entered the workplace and may take your job. [Електронний ресурс] : Сайт американського інформаційного каналу CNBC – Назва з екрану. Доступ: <https://www.cnbc.com/2019/10/31/human-like-androids-have-entered-the-workplace-and-may-take-your-job.html>
31. Frontiers | Soft Robots Manufacturing: A Review | Robotics and AI. [Електронний ресурс] : Відкрита наукова платформа – Назва з екрану. Доступ: <https://www.frontiersin.org/articles/10.3389/frobt.2018.00084/full>
32. Are you aware of these 7 challenges in robotics? | Artificial. [Електронний ресурс] : Сайт компанії Allerin – Назва з екрану. Доступ: <https://www.allerin.com/blog/are-you-aware-of-these-7-challenges-in-robotics>
33. Формальский А. М. Передвижение антропоморфных механизмов / А. М. Формальский. – М.: Наука. Главная редакция физико-математической литературы, 1982. – 369 с.
34. Тертычный-Даури В. Ю. Динамика робототехнических систем. Учебное пособие / В. Ю. Тертычный-Даури. – СПб.: НИУ ИТМО, 2012. – 128 с.
35. MATLAB and Simulink Robotics Arena: Walking Robot - File Exchange - MATLAB Central. [Електронний ресурс] : Файловий обмінник офіційного сайту MATLAB — Назва з екрану. Доступ: <https://www.mathworks.com/matlabcentral/fileexchange/64227-matlab-and-simulink-robotics-arena-walking-robot>
36. GitHub – mathworks/msra-walking-robot: Example files for MATLAB and Simulink Robotics Arena walking robot videos. [Електронний ресурс] : Веб-

ресурс для хостингу ІТ-проектів GitHub — Назва з екрану. Доступ:
<https://github.com/mathworks/msra-walking-robot>

37. Репнікова Н. Б. Вдосконалення функціональних можливостей моделі крокуючого робота / Н. Б. Репнікова, В. А. Сверчков // Проблеми інформатизації та управління: збірник наукових праць. – 2021. — Т. 1, №65 — С. 62-67. <https://doi.org/10.18372/2073-4751.65.15372>.
38. Bräunl T. Embedded Robotics: Mobile Robot Design and Application with Embedded Systems, 3rd Ed. / T. Bräunl — New York: Springer Publishing, 2008. — P. 165-184.
39. Замараев В. А., Година Е. З., Никитюк Д. Б. Анатомия для студентов физкультурных вузов и факультетов: учебник и практикум для академического бакалавриата / В. А. Замараев, Е. З. Година, Д. Б. Никитюк. — Москва: Издательство Юрайт, 2019. — 416 с.
40. Мощность и вращающий момент электродвигателя. Что это такое? [Электронный ресурс] : Інформаційна система ЕлектроТехІнфо – Назва з екрану. Доступ: https://eti.su/articles/elektricheskie-mashini/elektricheskie-mashini_1571.html
41. Yang T. Control of aperiodic walking and the energetic effects of parallel joint compliance of planar bipedal robots : дис. докт. філос. наук / Yang Tao — Columbus, Ohio, 2007. — 214 с.
42. Halloran M. Wing in Ground Effect Craft Review (DSTO-GD-0201). [Електронний ресурс] : М. Halloran, S. O`Meara // DSTO Aeronautical and Maritime Research Laboratory. — 1999. Доступ: <https://apps.dtic.mil/sti/pdfs/ADA361836.pdf>
43. Tucker V. A. The Energetic Cost of Moving About / V. A. Tucker — American scientists, v. 63(4):413-419, July-Aug. 1975.
44. Luneskas M. Hexapod Robot Gait Switching for Energy Consumption and Cost of Transport Management Using Heuristic Algorithms / M. Luneskas, T. Luneskas, J. Kriauciunas, D. Udris, D. Plonis D, R. Damaševicius, R.

Maskeliunas // Appl. Sci. – 2021. – Т. 11, Вып. 3. –
<https://doi.org/10.3390/app11031339>

45. Kim S. Design of Dynamic Legged Robots / S. Kim, P. M. Wensing // Now Foundations and Trends. – 2017. – Vol. 5, No. 2 (2014). – P. 117-190.
<https://doi.org/10.1561/23000000044>
46. Neville N. A bipedal running robot with one actuator per leg / N. Neville, M. Buehler, I. Sharf // 2006 IEEE International Conference on Robotics and Automation (ICRA), 2006. – pp. 848-853.
<https://doi.org/10.1109/ROBOT.2006.1641815>

ДОДАТОК А

Код обчислення швидкості моделі робота

```
array_data = yout{2}.Values.Data;
array_time = yout{2}.Values.Time;

len = length(array_data);
b = 1;

for a = 1:len
    if b == 1
        array_speed(b) = array_data(a);
    else
        array_speed(b) = (array_data(a) - array_data(a-1))/(array_time(a) -
array_time(a-1));
    end
    b = b+1;
end

result_speed = sum(array_speed)/len;

disp(length(yout{2}.Values.Data));
disp(length(array_speed));

% printing our array data
figure(13)
hold on
plot(yout{2}.Values.Time, array_speed);
xlabel('Time [s]');
ylabel('Robot`s speed [m/s]');

f = msgbox(sprintf('Average robot`s speed: %f m/s', result_speed));
```



```

power_array_0_gaitCycle_ankle_Z =
sum(abs(array_0_gaitCycle_ankle_Z))/length(array_0_gaitCycle_ankle_Z);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_ankle_Z = array_one_state(find(array_time_ankle_Z_velocity == 0.55)
: find(array_time_ankle_Z_velocity == gaitTime_sideWalking));
power_array_1_gaitCycle_ankle_Z =
sum(abs(array_1_gaitCycle_ankle_Z))/length(array_1_gaitCycle_ankle_Z);

% обчислення потужності під час усього часу ходьби
array_only_zero_ankle_Z = array_zero_state(find(array_time_ankle_Z_velocity == 0.55)
: end);
array_only_one_ankle_Z = array_one_state(find(array_time_ankle_Z_velocity == 0.55) :
end);

right_ankle_Z_power_walk = (length(array_only_zero_ankle_Z) *
power_array_0_gaitCycle_ankle_Z) / length(array_0_gaitCycle_ankle_Z);

right_ankle_Z_power_walk_side = (length(array_only_one_ankle_Z) *
power_array_1_gaitCycle_ankle_Z) / length(array_1_gaitCycle_ankle_Z);

full_right_ankle_Z_power_walk = right_ankle_Z_power_walk + 2*power_0_5_ankle_Z +
right_ankle_Z_power_walk_side;
% обчислення кількості циклів ходьби
walking_cycles_ankle_Z = length(array_only_zero_ankle_Z) /
length(array_0_gaitCycle_ankle_Z);
walking_cycles_ankle_Z_One_State = length(array_only_one_ankle_Z) /
length(array_1_gaitCycle_ankle_Z);

% виведення інформації
disp("*****");
disp("Right Ankle Z power before 0.5s is " + power_0_5_ankle_Z + " Watt");
disp("Right Ankle Z power from 0.55s to 1.3s is " + power_array_0_gaitCycle_ankle_Z +
" Watt");
disp("Energy consuption of Right Ankle Z during ZERO phase is " +
right_ankle_Z_power_walk + " Watt");
disp("Right Ankle Z power during side walk cycle: " + power_array_1_gaitCycle_ankle_Z
+ " Watt");
disp("Energy consuption of Right Ankle Z during ONE phase is " +
right_ankle_Z_power_walk_side + " Watt");
disp("FULL Energy consuption of Right Ankle Z is " + full_right_ankle_Z_power_walk +
" Watt");
disp("Walking cycles (zero): " + walking_cycles_ankle_Z);
disp("Walking cycles (one): " + walking_cycles_ankle_Z_One_State);

figure(31)
subplot(4,1,1)
hold on
plot(array_time_ankle_Z_velocity, array_data_ankle_Z_velocity);
xlabel('Time [s]');
ylabel('Ankle Z Velocity (Right) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_ankle_Z_velocity, array_data_ankle_Z_torque);
xlabel('Time [s]');
ylabel('Ankle Z Torque (Right) [N*m]');

subplot(4,1,3)
hold on

```

```

plot(array_time_ankle_Z_velocity, array_ankle_power);
xlabel('Time [s]');
ylabel('Ankle Z Power (Right) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_ankle_Z_velocity, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Hip Y (right)

% вивантажуємо необхідні дані

array_data_hip_Y_velocity = yout{7}.Values.hip_Y_velocity.Data;
array_time_hip_Y = yout{7}.Values.hip_Y_velocity.Time;

array_data_hip_Y_torque = yout{7}.Values.hip_Y_torque.Data;

% обчислення потужності
len_array_hip_Y = length(array_data_hip_Y_velocity);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_hip_Y
    temp = array_data_hip_Y_velocity(b) * array_data_hip_Y_torque(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_hip_Y_power(b) = temp;
    b = b+1;
end

array_states = yout{9}.Values.Data;

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_hip_Y
    if array_states(n) == 0
        array_zero_state(v) = array_hip_Y_power(n);
        v=v+1;
    else
        array_one_state(j) = array_hip_Y_power(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_hip_Y = array_zero_state(1 : find(array_time_hip_Y == 0.5));
power_0_5_hip_Y = sum(abs(array_0_5_hip_Y))/length(array_0_5_hip_Y);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_hip_Y = array_zero_state(find(array_time_hip_Y == 0.55) :
find(array_time_hip_Y == gaitTime_normalWalking));

```

```

power_array_0_gaitCycle_hip_Y =
sum(abs(array_0_gaitCycle_hip_Y))/length(array_0_gaitCycle_hip_Y);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_hip_Y = array_one_state(find(array_time_hip_Y == 0.55) :
find(array_time_hip_Y == gaitTime_sideWalking));
power_array_1_gaitCycle_hip_Y =
sum(abs(array_1_gaitCycle_hip_Y))/length(array_1_gaitCycle_hip_Y);

% обчислення потужності під час усього часу ходьби
array_only_zero_hip_Y = array_zero_state(find(array_time_hip_Y == 0.55) : end);
array_only_one_hip_Y = array_one_state(find(array_time_hip_Y == 0.55) : end);

right_hip_Y_power_walk = (length(array_only_zero_hip_Y) *
power_array_0_gaitCycle_hip_Y) / length(array_0_gaitCycle_hip_Y);

right_hip_Y_power_walk_side = (length(array_only_one_hip_Y) *
power_array_1_gaitCycle_hip_Y) / length(array_1_gaitCycle_hip_Y);

full_right_hip_Y_power_walk = right_hip_Y_power_walk + 2*power_0_5_hip_Y +
right_hip_Y_power_walk_side;
% обчислення кількості циклів ходьби
walking_cycles_hip_Y = length(array_only_zero_hip_Y) /
length(array_0_gaitCycle_hip_Y);
walking_cycles_hip_Y_One_State = length(array_only_one_hip_Y) /
length(array_1_gaitCycle_hip_Y);

% виведення інформації
disp("*****");
disp("Right Hip Y power before 0.5s is " + power_0_5_hip_Y + " Watt");
disp("Right Hip Y power from 0.55s to 1.3s is " + power_array_0_gaitCycle_hip_Y + "
Watt");
disp("Energy consuption of Right Hip Y during ZERO phase is " +
right_hip_Y_power_walk + " Watt");
disp("Right Hip Y power during side walk cycle: " + power_array_1_gaitCycle_hip_Y + "
Watt");
disp("Energy consuption of Right Hip Y during ONE phase is " +
right_hip_Y_power_walk_side + " Watt");
disp("FULL Energy consuption of Right Hip Y is " + full_right_hip_Y_power_walk + "
Watt");
disp("Walking cycles (zero): " + walking_cycles_hip_Y);
disp("Walking cycles (one): " + walking_cycles_hip_Y_One_State);

figure(32)
subplot(4,1,1)
hold on
plot(array_time_hip_Y, array_data_hip_Y_velocity);
xlabel('Time [s]');
ylabel('Hip Y Velocity (Right) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_hip_Y, array_data_hip_Y_torque);
xlabel('Time [s]');
ylabel('Hip Y Torque (Right) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_hip_Y, array_hip_Y_power);
xlabel('Time [s]');

```

```

ylabel('Hip Y Power (Right) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_hip_Y, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Ankle (right)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_ankle_velocity = yout{4}.Values.ankle_speed.Data;
array_time_ankle = yout{4}.Values.ankle_speed.Time;

array_data_ankle_torque = yout{4}.Values.ankle_torque.Data;

% обчислення потужності
len_array_ankle = length(array_data_ankle_velocity);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_ankle
    temp = array_data_ankle_velocity(b) * array_data_ankle_torque(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_ankle_power(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_ankle
    if array_states(n) == 0
        array_ankle_zero_state(v) = array_ankle_power(n);
        v=v+1;
    else
        array_ankle_one_state(j) = array_ankle_power(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_ankle = array_ankle_zero_state(1 : find(array_time_ankle == 0.5));
power_0_5_ankle = sum(abs(array_0_5_ankle))/length(array_0_5_ankle);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_ankle = array_ankle_zero_state(find(array_time_ankle == 0.55) :
find(array_time_ankle == gaitTime_normalWalking));
power_array_0_gaitCycle_ankle =
sum(abs(array_0_gaitCycle_ankle))/length(array_0_gaitCycle_ankle);

% обчислення потужності для перших 0.8 с при ходьбі боком

```

```

array_1_gaitCycle_ankle = array_ankle_one_state(find(array_time_ankle == 0.55) :
find(array_time_ankle == gaitTime_sideWalking));
power_array_1_gaitCycle_ankle =
sum(abs(array_1_gaitCycle_ankle))/length(array_1_gaitCycle_ankle);

% обчислення потужності під час усього часу ходьби
array_only_zero_ankle = array_ankle_zero_state(find(array_time_ankle == 0.55) : end);
array_only_one_ankle = array_ankle_one_state(find(array_time_ankle == 0.55) : end);

right_ankle_power_walk = (length(array_only_zero_ankle) *
power_array_0_gaitCycle_ankle) / length(array_0_gaitCycle_ankle);

right_ankle_power_walk_side = (length(array_only_one_ankle) *
power_array_1_gaitCycle_ankle) / length(array_1_gaitCycle_ankle);

full_right_ankle_power_walk = right_ankle_power_walk + 2*power_0_5_ankle +
right_ankle_power_walk_side;
% обчислення кількості циклів ходьби
walking_cycles_ankle = length(array_only_zero_ankle) /
length(array_0_gaitCycle_ankle);
walking_cycles_hip_One_State_ankle = length(array_only_one_ankle) /
length(array_1_gaitCycle_ankle);

% виведення інформації
disp("*****");
disp("Right Ankle power before 0.5s is " + power_0_5_ankle + " Watt");
disp("Right Ankle power from 0.55s to 1.3s is " + power_array_0_gaitCycle_ankle + "
Watt");
disp("Energy consuption of Right Ankle during ZERO phase is " +
right_ankle_power_walk + " Watt");
disp("Right Ankle power during side walk cycle: " + power_array_1_gaitCycle_ankle + "
Watt");
disp("Energy consuption of Right Ankle during ONE phase is " +
right_ankle_power_walk_side + " Watt");
disp("FULL Energy consuption of Right Ankle is " + full_right_ankle_power_walk + "
Watt");
disp("Walking cycles (zero): " + walking_cycles_ankle);
disp("Walking cycles (one): " + walking_cycles_hip_One_State_ankle);

figure(33)
subplot(4,1,1)
hold on
plot(array_time_ankle, array_data_ankle_velocity);
xlabel('Time [s]');
ylabel('Ankle Velocity (Right) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_ankle, array_data_ankle_torque);
xlabel('Time [s]');
ylabel('Ankle Torque (Right) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_ankle, array_ankle_power);
xlabel('Time [s]');
ylabel('Ankle Power (Right) [Watt]');

subplot(4,1,4)
hold on

```

```

plot(array_time_ankle, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Knee (right)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_knee_velocity = yout{4}.Values.knee_speed.Data;
array_time_knee = yout{4}.Values.knee_speed.Time;

array_data_knee_torque = yout{4}.Values.knee_torque.Data;

% обчислення потужності
len_array_knee = length(array_data_knee_velocity);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_knee
    temp = array_data_knee_velocity(b) * array_data_knee_torque(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_knee_power(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_knee
    if array_states(n) == 0
        array_knee_zero_state(v) = array_knee_power(n);
        v=v+1;
    else
        array_knee_one_state(j) = array_knee_power(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_knee = array_knee_zero_state(1 : find(array_time_knee == 0.5));
power_0_5_knee = sum(abs(array_0_5_knee))/length(array_0_5_knee);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_knee = array_knee_zero_state(find(array_time_knee == 0.55) :
find(array_time_knee == gaitTime_normalWalking));
power_array_0_gaitCycle_knee =
sum(abs(array_0_gaitCycle_knee))/length(array_0_gaitCycle_knee);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_knee = array_knee_one_state(find(array_time_knee == 0.55) :
find(array_time_knee == gaitTime_sideWalking));
power_array_1_gaitCycle_knee =
sum(abs(array_1_gaitCycle_knee))/length(array_1_gaitCycle_knee);

```



```

% обчислення потужності під час усього часу ходьби
array_only_zero_knee = array_knee_zero_state(find(array_time_knee == 0.55) : end);
array_only_one_knee = array_knee_one_state(find(array_time_knee == 0.55) : end);

right_knee_power_walk = (length(array_only_zero_knee) * power_array_0_gaitCycle_knee)
/ length(array_0_gaitCycle_knee);

right_knee_power_walk_side = (length(array_only_one_knee) *
power_array_1_gaitCycle_knee) / length(array_1_gaitCycle_knee);

full_right_knee_power_walk = right_knee_power_walk + power_0_5_knee +
right_knee_power_walk_side;
% обчислення кількості циклів ходьби
walking_cycles_knee = length(array_only_zero_knee) / length(array_0_gaitCycle_knee);
walking_cycles_knee_One_State = length(array_only_one_knee) /
length(array_1_gaitCycle_knee);

% виведення інформації
disp("*****");
disp("Right Knee power before 0.5s is " + power_0_5_knee + " Watt");
disp("Right Knee power from 0.55s to 1.3s is " + power_array_0_gaitCycle_knee + "
Watt");
disp("Energy consuption of Right Knee during ZERO phase is " + right_knee_power_walk
+ " Watt");
disp("Right Knee power during side walk cycle: " + power_array_1_gaitCycle_knee + "
Watt");
disp("Energy consuption of Right Knee during ONE phase is " +
right_knee_power_walk_side + " Watt");
disp("FULL Energy consuption of Right Knee is " + full_right_knee_power_walk + "
Watt");
disp("Walking cycles (zero): " + walking_cycles_knee);
disp("Walking cycles (one): " + walking_cycles_knee_One_State);

figure(34)
subplot(4,1,1)
hold on
plot(array_time_knee, array_data_knee_velocity);
xlabel('Time [s]');
ylabel('Knee Velocity (Right) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_knee, array_data_knee_torque);
xlabel('Time [s]');
ylabel('Knee Torque (Right) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_knee, array_knee_power);
xlabel('Time [s]');
ylabel('Knee Power (Right) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_knee, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

```

```

%% evaluations for Hip (right)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_hip_right_velocity = yout{4}.Values.hip_speed.Data;
array_time_hip_right = yout{4}.Values.hip_speed.Time;

array_data_hip_right_torque = yout{4}.Values.hip_torque.Data;

% обчислення потужності
len_array_hip_right = length(array_data_hip_right_velocity);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_hip_right
    temp = array_data_hip_right_velocity(b) * array_data_hip_right_torque(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_hip_right_power(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_hip_right
    if array_states(n) == 0
        array_hip_right_zero_state(v) = array_hip_right_power(n);
        v=v+1;
    else
        array_hip_right_one_state(j) = array_hip_right_power(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_hip_right = array_hip_right_zero_state(1 : find(array_time_hip_right == 0.5));
power_0_5_hip_right = sum(abs(array_0_5_hip_right))/length(array_0_5_hip_right);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_hip_right = array_hip_right_zero_state(find(array_time_hip_right == 0.55) : find(array_time_hip_right == gaitTime_normalWalking));
power_array_0_gaitCycle_hip_right = sum(abs(array_0_gaitCycle_hip_right))/length(array_0_gaitCycle_hip_right);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_hip_right = array_hip_right_one_state(find(array_time_hip_right == 0.55) : find(array_time_hip_right == gaitTime_sideWalking));
power_array_1_gaitCycle_hip_right = sum(abs(array_1_gaitCycle_hip_right))/length(array_1_gaitCycle_hip_right);

% обчислення потужності під час усього часу ходьби
array_only_zero_hip_right = array_hip_right_zero_state(find(array_time_hip_right == 0.55) : end);
array_only_one_hip_right = array_hip_right_one_state(find(array_time_hip_right == 0.55) : end);

```

```

right_hip_right_power_walk = (length(array_only_zero_hip_right) *
power_array_0_gaitCycle_hip_right) / length(array_0_gaitCycle_hip_right);

right_hip_right_power_walk_side = (length(array_only_one_hip_right) *
power_array_1_gaitCycle_hip_right) / length(array_1_gaitCycle_hip_right);

full_right_hip_right_power_walk = right_hip_right_power_walk + 2*power_0_5_hip_right
+ right_hip_right_power_walk_side;

% виведення інформації
disp("*****");
disp("Right Hip power before 0.5s is " + power_0_5_hip_right + " Watt");
disp("Right Hip power from 0.55s to 1.3s is " + power_array_0_gaitCycle_hip_right + "
Watt");
disp("Energy consuption of Right Hip during ZERO phase is " +
right_hip_right_power_walk + " Watt");
disp("Right Hip power during side walk cycle: " + power_array_1_gaitCycle_hip_right +
" Watt");
disp("Energy consuption of Right Hip during ONE phase is " +
right_hip_right_power_walk_side + " Watt");
disp("FULL Energy consuption of Right Hip is " + full_right_hip_right_power_walk + "
Watt");

figure(35)
subplot(4,1,1)
hold on
plot(array_time_hip_right, array_data_hip_right_velocity);
xlabel('Time [s]');
ylabel('Hip Velocity (Right) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_hip_right, array_data_hip_right_torque);
xlabel('Time [s]');
ylabel('Hip Torque (Right) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_hip_right, array_hip_right_power);
xlabel('Time [s]');
ylabel('Hip Power (Right) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_hip_right, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Hip Y (left)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_hip_Y_velocity_l = yout{8}.Values.hip_Y_velocity.Data;
array_time_hip_Y_l = yout{8}.Values.hip_Y_velocity.Time;

array_data_hip_Y_torque_l = yout{8}.Values.hip_Y_torque.Data;

```

```

% обчислення потужності
len_array_hip_Y_1 = length(array_data_hip_Y_velocity_1);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_hip_Y_1
    temp = array_data_hip_Y_velocity_1(b) * array_data_hip_Y_torque_1(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_hip_Y_power_1(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_hip_Y_1
    if array_states(n) == 0
        array_hip_Y_zero_state_1(v) = array_hip_Y_power_1(n);
        v=v+1;
    else
        array_hip_Y_one_state_1(j) = array_hip_Y_power_1(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_hip_Y_1 = array_hip_Y_zero_state_1(1 : find(array_time_hip_Y_1 == 0.5));
power_0_5_hip_Y_1 = sum(abs(array_0_5_hip_Y_1))/length(array_0_5_hip_Y_1);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_hip_Y_1 = array_hip_Y_zero_state_1(find(array_time_hip_Y_1 == 0.55) : find(array_time_hip_Y_1 == gaitTime_normalWalking));
power_array_0_gaitCycle_hip_Y_1 = sum(abs(array_0_gaitCycle_hip_Y_1))/length(array_0_gaitCycle_hip_Y_1);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_hip_Y_1 = array_hip_Y_one_state_1(find(array_time_hip_Y_1 == 0.55) : find(array_time_hip_Y_1 == gaitTime_sideWalking));
power_array_1_gaitCycle_hip_Y_1 = sum(abs(array_1_gaitCycle_hip_Y_1))/length(array_1_gaitCycle_hip_Y_1);

% обчислення потужності під час усього часу ходьби
array_only_zero_hip_Y_1 = array_hip_Y_zero_state_1(find(array_time_hip_Y_1 == 0.55) : end);
array_only_one_hip_Y_1 = array_hip_Y_one_state_1(find(array_time_hip_Y_1 == 0.55) : end);

right_hip_Y_power_walk_1 = (length(array_only_zero_hip_Y_1) * power_array_0_gaitCycle_hip_Y_1) / length(array_0_gaitCycle_hip_Y_1);

right_hip_Y_power_walk_side_1 = (length(array_only_one_hip_Y_1) * power_array_1_gaitCycle_hip_Y_1) / length(array_1_gaitCycle_hip_Y_1);

```

```

full_right_hip_Y_power_walk_l = right_hip_Y_power_walk_l + 2*power_0_5_hip_Y_l +
right_hip_Y_power_walk_side_l;

% виведення інформації
disp("*****");
disp("Left Hip Y power before 0.5s is " + power_0_5_hip_Y_l + " Watt");
disp("Left Hip Y power from 0.55s to 1.3s is " + power_array_0_gaitCycle_hip_Y_l + "
Watt");
disp("Energy consumption of Left Hip Y during ZERO phase is " +
right_hip_Y_power_walk_l + " Watt");
disp("Left Hip Y power during side walk cycle: " + power_array_1_gaitCycle_hip_Y_l +
" Watt");
disp("Energy consumption of Left Hip Y during ONE phase is " +
right_hip_Y_power_walk_side_l + " Watt");
disp("FULL Energy consumption of Left Hip Y is " + full_right_hip_Y_power_walk_l + "
Watt");

figure(36)
subplot(4,1,1)
hold on
plot(array_time_hip_Y_l, array_data_hip_Y_velocity_l);
xlabel('Time [s]');
ylabel('Hip Y Velocity (Left) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_hip_Y_l, array_data_hip_Y_torque_l);
xlabel('Time [s]');
ylabel('Hip Y Torque (Left) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_hip_Y_l, array_hip_Y_power_l);
xlabel('Time [s]');
ylabel('Hip Y Power (Left) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_hip_Y_l, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Ankle (left)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_ankle_velocity_l = yout{5}.Values.ankle_speed.Data;
array_time_ankle_l = yout{5}.Values.ankle_speed.Time;

array_data_ankle_torque_l = yout{5}.Values.ankle_torque.Data;

% обчислення потужності
len_array_ankle_l = length(array_data_ankle_velocity_l);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_ankle_l
    temp = array_data_ankle_velocity_l(b) * array_data_ankle_torque_l(b);

```

```

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_ankle_power_1(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_ankle_1
    if array_states(n) == 0
        array_ankle_zero_state_1(v) = array_ankle_power_1(n);
        v=v+1;
    else
        array_ankle_one_state_1(j) = array_ankle_power_1(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_ankle_1 = array_ankle_zero_state_1(1 : find(array_time_ankle_1 == 0.5));
power_0_5_ankle_1 = sum(abs(array_0_5_ankle_1))/length(array_0_5_ankle_1);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_ankle_1 = array_ankle_zero_state_1(find(array_time_ankle_1 == 0.55)
: find(array_time_ankle_1 == gaitTime_normalWalking));
power_array_0_gaitCycle_ankle_1 =
sum(abs(array_0_gaitCycle_ankle_1))/length(array_0_gaitCycle_ankle_1);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_ankle_1 = array_ankle_one_state_1(find(array_time_ankle_1 == 0.55)
: find(array_time_ankle_1 == gaitTime_sideWalking));
power_array_1_gaitCycle_ankle_1 =
sum(abs(array_1_gaitCycle_ankle_1))/length(array_1_gaitCycle_ankle_1);

% обчислення потужності під час усього часу ходьби
array_only_zero_ankle_1 = array_ankle_zero_state_1(find(array_time_ankle_1 == 0.55) :
end);
array_only_one_ankle_1 = array_ankle_one_state_1(find(array_time_ankle_1 == 0.55) :
end);

left_ankle_power_walk = (length(array_only_zero_ankle_1) *
power_array_0_gaitCycle_ankle_1) / length(array_0_gaitCycle_ankle_1);

left_ankle_power_walk_side = (length(array_only_one_ankle_1) *
power_array_1_gaitCycle_ankle_1) / length(array_1_gaitCycle_ankle_1);

full_left_ankle_power_walk = left_ankle_power_walk + 2*power_0_5_ankle_1 +
left_ankle_power_walk_side;

% виведення інформації
disp("*****");
disp("Left Ankle power before 0.5s is " + power_0_5_ankle_1 + " Watt");
disp("Left Ankle power from 0.55s to 1.3s is " + power_array_0_gaitCycle_ankle_1 + "
Watt");
disp("Energy consumption of Left Ankle during ZERO phase is " + left_ankle_power_walk
+ " Watt");

```

```

disp("Left Ankle power during side walk cycle: " + power_array_1_gaitCycle_ankle_1 +
" Watt");
disp("Energy consuption of Left Ankle during ONE phase is " +
left_ankle_power_walk_side + " Watt");
disp("FULL Energy consuption of Left Ankle is " + full_left_ankle_power_walk + "
Watt");

figure(37)
subplot(4,1,1)
hold on
plot(array_time_ankle_1, array_data_ankle_velocity_1);
xlabel('Time [s]');
ylabel('Ankle Velocity (Left) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_ankle_1, array_data_ankle_torque_1);
xlabel('Time [s]');
ylabel('Ankle Torque (Left) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_ankle_1, array_ankle_power_1);
xlabel('Time [s]');
ylabel('Ankle Power (Left) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_ankle_1, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Knee (left)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_knee_velocity_1 = yout{5}.Values.knee_speed.Data;
array_time_knee_1 = yout{5}.Values.knee_speed.Time;

array_data_knee_torque_1 = yout{5}.Values.knee_torque.Data;

% обчислення потужності
len_array_knee_1 = length(array_data_knee_velocity_1);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_knee_1
    temp = array_data_knee_velocity_1(b) * array_data_knee_torque_1(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_knee_power_1(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)

```

```

v = 1;
j = 1;
for n = 1:len_array_knee_1
    if array_states(n) == 0
        array_knee_zero_state_1(v) = array_knee_power_1(n);
        v=v+1;
    else
        array_knee_one_state_1(j) = array_knee_power_1(n);
        j=j+1;
    end
end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_knee_1 = array_knee_zero_state_1(1 : find(array_time_knee_1 == 0.5));
power_0_5_knee_1 = sum(abs(array_0_5_knee_1))/length(array_0_5_knee_1);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_knee_1 = array_knee_zero_state_1(find(array_time_knee_1 == 0.55) :
find(array_time_knee_1 == gaitTime_normalWalking));
power_array_0_gaitCycle_knee_1 =
sum(abs(array_0_gaitCycle_knee_1))/length(array_0_gaitCycle_knee_1);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_knee_1 = array_knee_one_state_1(find(array_time_knee_1 == 0.55) :
find(array_time_knee_1 == gaitTime_sideWalking));
power_array_1_gaitCycle_knee_1 =
sum(abs(array_1_gaitCycle_knee_1))/length(array_1_gaitCycle_knee_1);

% обчислення потужності під час усього часу ходьби
array_only_zero_knee_1 = array_knee_zero_state_1(find(array_time_knee_1 == 0.55) :
end);
array_only_one_knee_1 = array_knee_one_state_1(find(array_time_knee_1 == 0.55) :
end);

left_knee_power_walk = (length(array_only_zero_knee_1) *
power_array_0_gaitCycle_knee_1) / length(array_0_gaitCycle_knee_1);

left_knee_power_walk_side = (length(array_only_one_knee_1) *
power_array_1_gaitCycle_knee_1) / length(array_1_gaitCycle_knee_1);

full_left_knee_power_walk = left_knee_power_walk + power_0_5_knee_1 +
left_knee_power_walk_side;

% виведення інформації
disp("*****");
disp("Left Knee power before 0.5s is " + power_0_5_knee_1 + " Watt");
disp("Left Knee power from 0.55s to 1.3s is " + power_array_0_gaitCycle_knee_1 + "
Watt");
disp("Energy consuption of Left Knee during ZERO phase is " + left_knee_power_walk +
" Watt");
disp("Left Knee power during side walk cycle: " + power_array_1_gaitCycle_knee_1 + "
Watt");
disp("Energy consuption of Left Knee during ONE phase is " +
left_knee_power_walk_side + " Watt");
disp("FULL Energy consuption of Left Knee is " + full_left_knee_power_walk + "
Watt");

figure(38)
subplot(4,1,1)
hold on

```



```

plot(array_time_knee_1, array_data_knee_velocity_1);
xlabel('Time [s]');
ylabel('Knee Velocity (Left) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_knee_1, array_data_knee_torque_1);
xlabel('Time [s]');
ylabel('Knee Torque (Left) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_knee_1, array_knee_power_1);
xlabel('Time [s]');
ylabel('Knee Power (Left) [Watt]');

subplot(4,1,4)
hold on
plot(array_time_knee_1, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

%% evaluations for Hip (left)

% вивантажуємо необхідні дані
array_states = yout{9}.Values.Data;
array_data_hip_velocity_1 = yout{5}.Values.hip_speed.Data;
array_time_hip_1 = yout{5}.Values.hip_speed.Time;

array_data_hip_torque_1 = yout{5}.Values.hip_torque.Data;

% обчислення потужності
len_array_hip_1 = length(array_data_hip_velocity_1);
b = 1;
a = 1;
temp = 0;

for a = 1:len_array_hip_1
    temp = array_data_hip_velocity_1(b) * array_data_hip_torque_1(b);

    if temp > 500 || abs(temp) > 500
        temp = 500;
    end

    array_hip_power_1(b) = temp;
    b = b+1;
end

% вивантаження станів з 0 (робот йде) та з 1 (робот обходить перешкоду)
v = 1;
j = 1;
for n = 1:len_array_hip_1
    if array_states(n) == 0
        array_hip_zero_state_1(v) = array_hip_power_1(n);
        v=v+1;
    else
        array_hip_one_state_1(j) = array_hip_power_1(n);
        j=j+1;
    end
end

```

```

end

% обчислення потужності для перших 0.5 с при прямолінійній ходьбі
array_0_5_hip_1 = array_hip_zero_state_1(1 : find(array_time_hip_1 == 0.5));
power_0_5_hip_1 = sum(abs(array_0_5_hip_1))/length(array_0_5_hip_1);

% обчислення потужності під час циклу ходьби
array_0_gaitCycle_hip_1 = array_hip_zero_state_1(find(array_time_hip_1 == 0.55) :
find(array_time_hip_1 == gaitTime_normalWalking));
power_array_0_gaitCycle_hip_1 =
sum(abs(array_0_gaitCycle_hip_1))/length(array_0_gaitCycle_hip_1);

% обчислення потужності для перших 0.8 с при ходьбі боком
array_1_gaitCycle_hip_1 = array_hip_one_state_1(find(array_time_hip_1 == 0.55) :
find(array_time_hip_1 == gaitTime_sideWalking));
power_array_1_gaitCycle_hip_1 =
sum(abs(array_1_gaitCycle_hip_1))/length(array_1_gaitCycle_hip_1);

% обчислення потужності під час усього часу ходьби
array_only_zero_hip_1 = array_hip_zero_state_1(find(array_time_hip_1 == 0.55) : end);
array_only_one_hip_1 = array_hip_one_state_1(find(array_time_hip_1 == 0.55) : end);

left_hip_power_walk = (length(array_only_zero_hip_1) * power_array_0_gaitCycle_hip_1)
/ length(array_0_gaitCycle_hip_1);

left_hip_power_walk_side = (length(array_only_one_hip_1) *
power_array_1_gaitCycle_hip_1) / length(array_1_gaitCycle_hip_1);

full_left_hip_power_walk = left_hip_power_walk + 2*power_0_5_hip_1 +
left_hip_power_walk_side;

% виведення інформації
disp("*****");
disp("Left Hip power before 0.5s is " + power_0_5_hip_1 + " Watt");
disp("Left Hip power from 0.55s to 1.3s is " + power_array_0_gaitCycle_hip_1 + "
Watt");
disp("Energy consuption of Left Hip during ZERO phase is " + left_hip_power_walk + "
Watt");
disp("Left Hip power during side walk cycle: " + power_array_1_gaitCycle_hip_1 + "
Watt");
disp("Energy consuption of Left Hip during ONE phase is " + left_hip_power_walk_side
+ " Watt");
disp("FULL Energy consuption of Left Hip is " + full_left_hip_power_walk + " Watt");

figure(39)
subplot(4,1,1)
hold on
plot(array_time_hip_1, array_data_hip_velocity_1);
xlabel('Time [s]');
ylabel('Hip Velocity (Left) [rad/s]');

subplot(4,1,2)
hold on
plot(array_time_hip_1, array_data_hip_torque_1);
xlabel('Time [s]');
ylabel('Hip Torque (Left) [N*m]');

subplot(4,1,3)
hold on
plot(array_time_hip_1, array_hip_power_1);

```

```

xlabel('Time [s]');
ylabel('Hip Power (Left) [Watt]');

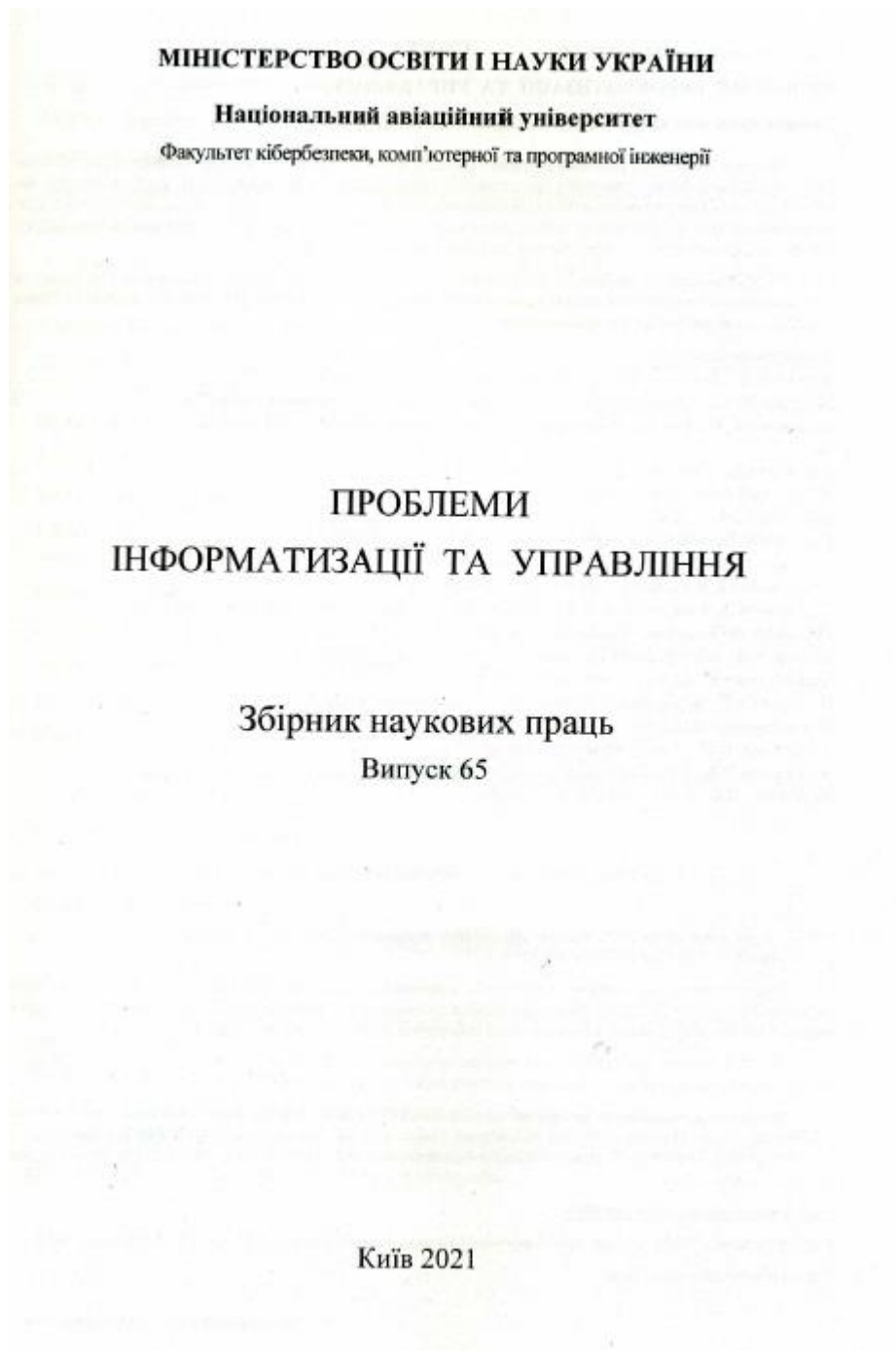
subplot(4,1,4)
hold on
plot(array_time_hip_l, array_states);
xlabel('Time [s]');
ylabel('States of Robot [Watt]');

disp("MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM");
sum_P = full_right_ankle_Z_power_walk + full_right_hip_Y_power_walk + ...
full_right_ankle_power_walk + full_right_knee_power_walk + ...
full_right_hip_right_power_walk + full_right_hip_Y_power_walk_l + ...
full_left_ankle_power_walk + full_left_knee_power_walk + full_left_hip_power_walk;
disp ("Sum of all powers is " + sum_P + " Watt");

```

ДОДАТОК В

Перелік опублікованих матеріалів



УДК 004 – 047.64:378.4 (477-25)

ПРОБЛЕМИ ІНФОРМАТИЗАЦІЇ ТА УПРАВЛІННЯ:

Збірник наукових праць: Випуск 65 (1). – К.: НАУ, 2021. – 98 с.

Збірник охоплює проблеми аналізу і синтезу комп'ютерно-телекомунікаційних систем та систем управління ними: створення ефективного інформаційно-вимірювального інструментарію визначення та оцінки параметрів об'єктів інформаційних технологій; математичне, лінгвістичне, апаратно-програмне забезпечення нових, включаючи кіберфізичні авіаційні, і вдосконалених існуючих розподілених комп'ютеризованих та комп'ютерних систем.

Розрахований на наукових працівників та фахівців, які займаються питаннями створення та вдосконалення теоретичної та програмно-технічної бази засобів комп'ютерно-телекомунікаційних систем та їх комп'ютерних компонентів.

Редакційна колегія:

Жуков І.А., д.т.н., професор, Україна (головний редактор)

Печурін М.К., д.т.н., професор, Україна. (заступник головного редактора)

Кудренко С.О., к.т.н., Україна (відповідальний секретар)

Азаров О.Д., д.т.н., професор, Україна

Білецький В.М., д.т.н., професор, Польща

Воронін А.М., д.т.н., професор, Україна

Гамаюн В.П., д.т.н., професор, Україна

Головко В.А., д.т.н., професор, Білорусь

Козловський В.В., д.т.н., професор, Україна

Меліков А.З. огли, чл.-кор. НАН Азербайджана, д.т.н., професор, Азербайджан

Мельник А.О., д.т.н., професор, Україна

Мохор В.В., чл.-кор. НАН України, д.т.н., професор, Україна

Опанасенко В.М., д.т.н., професор, Україна

Палагін О.В., академік НАН України, д.т.н., професор, Україна

Пономаренко Л.А., д.т.н., професор, Україна

Тарасенко В.П., д.т.н., професор, Україна

Хамдамов Р.Х., д.т.н., професор, Узбекистан

Харченко В.С., д.т.н., професор, Україна

Рекомендовано до друку Вченою радою Національного авіаційного університету (протокол № 3 від 24 березня 2021 р.)

Зареєстрований державним комітетом телебачення та радіомовлення України. Свідоцтво про реєстрацію № 8280 від 29.12.2003 р. Редакція не обов'язково поділяє думку автора. Відповідальність за достовірність фактів, цитат, власних імен та іншої інформації несуть автори публікацій.

Збірник входить до Переліку наукових фахових видань МОН України (категорія «Б»), у яких можуть публікуватися основні результати дисертаційних робіт з технічних наук.

Реферати та індексація: Google Scholar; eLIBRARY (RSCI); Національна бібліотека України імені В.І.Вернадського; Науково-технічна бібліотека Національного авіаційного університету; Simle Search Metadata (SSM); WorldCat (OAIster); Bielefeld Academic Search Engine (BASE); DRIVER; Ulrichsweb global serials directory.

<http://jrn1.nau.edu.ua/index.php/PIU/>

Адреса редакції: 03680, м. Київ, пр-т Любомира Гузара, 1, корпус 5, кім. 116, тел. (+ 38-044) 406-76-78.

E-mail: sbornik.piu@gmail.com

ЗМІСТ

Азарова А.О., Біліченко Н.О.	
Гібридний засіб захисту мовної інформації.....	4
Антонішин М.В., Дорогий Я.Ю., Міснік О.І., Цуркан В.В.	
Пошук реальних мережевих адрес веб застосунків використанням сервісу CDN...	10
Исмаилов Б.Г.	
Анализ системы безопасности информации с неограниченным объемом буферной памяти в сетях обслуживания.....	18
Касумов В.А., Алиева К.Дж.	
Достоверность комплексных индексов для флуоресцентно-гиперспектрального метода оценки содержания хлорофилла в листьях растений.....	25
Касумов В.А., Амашов Я.А., Исмаилов Дж.М.	
Интегрирование технологий IoT и FOG computing.....	31
Климова А.С., Савченко А.С., Кулик С.В.	
Параметричний синтез аналітико-психологічних систем на основі методів математичного моделювання та багатокритеріальної оптимізації.....	39
Kovalenko Yu.B.	
Experiment test of the reliability of computer systems of integrated modular avionics...	45
Kudrenko S.A., Zhuravel C.V., Fomina N.B.	
Overview and justification for choosing technology stack for data analysis system.....	54
Мехбальева Э.В.	
Схема определения скачкообразных приоритетов в системах обслуживания с гетерогенными серверами.....	62
Репнікова Н.Б., Сверчков В.А.	
Вдосконалення функціональних можливостей моделі крокуючого робота.....	68
Сушко С.В., Чемерис О.А.	
Інтелектуальний метод розбиття ітераційного простору оператору циклів програм	75
Терехов В.Є., Подгуренко В.С.	
Моделювання кривих потужності промислових вітроелектричних установок заданого типорозміру.....	83
Цитовцева А.С., Сопов О.О.	
Аналіз доступності мікросервісів на базі системи управління та оркестрації контейнерів KUBERNETES.....	91
Правила оформлення і представлення публікацій	97

ВДОСКОНАЛЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ МОДЕЛІ КРОКУЮЧОГО РОБОТА

Національний технічний університет України «Київський політехнічний інститут
ім. Ігоря Сікорського»

n.repnikova@gmail.com
 vladenigmatic@gmail.com

Вступ

Двоногі крокуючі роботи є досить цінними різновидами роботів, оскільки вони за своєю конструкцією правдоподібно відтворюють нижні кінцівки людини. Ця особливість відкриває перед науковцями широкі можливості для дослідження механіки руху нижніх кінцівок людини, а також допомагає проектувати пристрої, котрі виконували б певні транспортні задачі, асистували людині у небезпечному середовищі, або навіть допомагали хворим з паралізованістю ніг в їхній реабілітації [1-2].

Багато природніх ландшафтів мають досить нерівну поверхню: гори, пагорби, узбережжя тощо. Це дуже погані умови для використання колісних та навіть гусеничних роботів. Така ситуація підвищує затребуваність саме ходячих роботів з декількома ногами та їх актуальність для дослідників, оскільки вони відтворюють динаміку руху живих істот, котрі вдало переміщуються та живуть у подібних місцевостях [3].

Звичайно, розробка, вивчення та експериментування над такими прототипами неможливі без попередніх розрахунків або, принаймні, ескізів. На сьогодні є велика кількість доступного програмного забезпечення, котре допомагає виконувати моделювання різноманітних механізмів – в тому числі і роботів – без зайвих ресурсних та часових затрат. Одним з таких програмних забезпечень є пакет прикладних

програм MATLAB разом з інтерактивним інструментом Simulink.

В даній статті для досліджень була взята вже готова авторська модель, яку розробив Sebastian Castro (надалі – «автор») у середовищі MATLAB R2017a [4]. Запропонована автором модель реалізована з використанням спеціальної бібліотеки Simscape Multibody, яка входить до пакету Simulink, і котра надає можливість моделювати різноманітні механічні 3D системи.

Авторська модель являє собою простий двоногий крокуючий робот із шістьма степенями свободи: дві для стоп, дві для гомілкових сегментів, дві для стегнових сегментів. Він має висоту 28 см, вага складає 0.6273 кг, здатний відтворювати рухи нижніх кінцівок людини та ходити прямолінійно відповідно до заданого початкового напрямку. Авторська модель крокуючого робота є гарною основою для її подальшої модернізації під задачу виявлення та обходження перешкод на шляху робота.

Найбільш лаконічно авторський робот зображений на рис.1.

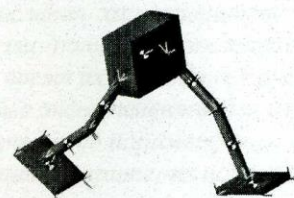


Рис. 1. Авторський крокуючий двоногий робот у середовищі Simulink

На рис. 1 чорно-білі кола – центри мас блоків, зеленим кольором позначено вісь Y (відповідає за ширину складових робота та ширину дороги), червоним – вісь X (відповідає за довжину дороги та складових робота), синім – вісь Z (відповідає за висоту). Маючи шість степенів свободи, робот має можливість здійснювати згинання стопи, коліна і стегна на лівій нозі та на правій. Це забезпечується можливістю відхилення кожного з цих трьох сегментів ноги відносно осі X на 180° – від -90° до 90° відносно початкової позиції з використанням блоків Rigid Transform бібліотеки Simscape Multibody.

Стопи робота мають у чотирьох кутах реалізовані контактні сили, котрі визначають взаємодію стоп робота з поверхнею, опираючись на яку він рухається.

Робот рухається за рахунок подання кожної 0.1333 секунди значень кутів на приводи стоп, колін та стегон з використанням блоків Repeating Table. Повний цикл ходьби робота триває 0.8 секунд, після чого він повторюється, доки робот з якихось причин не впаде, або доки час моделювання не припиниться. Спочатку значення кутів подаються на праву ногу, а через 0.4 секунди на ліву, що дозволяє ногам рухатися по чергово, виконуючи рівномірне перенесення ваги. За перетворення прийнятих значень кутів у відповідні керуючі сигнали для запуску виконавчих механізмів відповідає Simulink-блок Revolute Joint. Процес ходьби авторського робота зображено на рис 2.

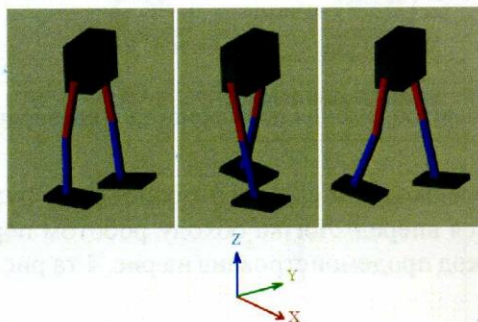


Рис. 2. Ходьба авторського робота у Simulink

Також, автор реалізував виведення інформації щодо певних параметрів робота у вигляді графіків: залежність кута стоп, колін та стегон від часу, пройдена роботою відстань, кутова швидкість, висота торсу робота над поверхнею.

Головна задача, котру вирішує авторська модель – переміщення робота у просторі подібно до паттерну переміщення людини. Мета модернізації моделі полягає у підвищенні функціональних можливостей робота, а саме – у наданні роботом можливості обходити перешкоди за їх наявності.

Шляхи та результати модернізації моделі крокуючого робота

На шляху до досягнення мети модернізації були виконані наступні кроки:

1. Зміна розрахунку інерції частин робота зі щільності на кілограми. До цього інерція сегментів нижніх кінцівок робота розраховувалася на основі щільності ($\text{кг}/\text{м}^3$), що було незручно для обчислення такого параметру робота, як вага. Також, відомо, що двоногих роботів досить важко зрівноважити у порівнянні з тими самими триногими або шестиногими екземплярами [5]. Вага модернізованого робота складає 172 кг. Така величина зумовлена тим, що робот має досить важкий торс у порівнянні з масою ніг, а тому збереження малої ваги ніг при переміщенні викликало б різку втрату рівноваги робота, що неминуче призводило б до його падіння. Для компенсації такого явища вага кожної стопи була прирівняна до 60 кг, при цьому гомілки та стегна – по 8 кг. Якщо казати про втілення даної моделі в реальності, то немає необхідності створювати такі важкі стопи – це лише найпростіша варіація усунення проблеми високої інерційності робота при віртуальному моделюванні, а у житті різні, наприклад, гідравлічні механізми можуть утримувати баланс робота.

2. Зміна розмірів робота на такі, що наближені до людини. Нові розміри складових робота (висота, вісь Z): стегна – 38 см, гомілки – 40 см, торс – 32 см. Дані розміри стегон та гомілок відповідають людині, чий зріст дорівнює 175 см.

3. Зміна параметрів взаємодії робота з поверхнею (вище згадані контактні сили).

4. Реалізація додаткової мобільності робота за рахунок впровадження додаткових 6-ти степенів свободи робота шляхом додавання можливості відхилення стегон відносно вісі Y, відхилення стоп відносно осей Y та Z. Результуюча модель має 12 степенів свободи. Таким чином, модернізована модель містить більш наближену до людини кількість степенів свободи [6].

5. Написання скрипту, котрий показує швидкість руху робота у кожний момент часу, а також підраховує середню швидкість робота.

6. Додавання перешкоди на шляху робота – паралелепіпед з розмірами $40 \times 80 \times 16$ см.

7. Реалізація виявлення роботом перешкоди та її уникнення зі збереженням

початкового напрямку руху (вздовж осі X).

Зупинимось детальніше на 7-му пункті. Його ядром є два блоки – Transform Sensor та Compare to Constant. Перший блок дозволяє аналізувати відстань між роботом та перешкодою, а другий – при виявленні достатнього зближення із перешкодою (тут – 1.3 м) перейти до відповідного реагування, яке детальніше буде розглядатися.

На рис. 3 зображена ділянка схеми, котра відповідає за просторове відображення перешкоди та її виявлення модернізованим роботом. Червоним виділенням позначена перешкода та чотири блоки, котрі відповідають за її просторове положення. Синім позначена ділянка, що аналізує відстань між роботом та перешкодою, а зеленим – детектор близькості до перешкоди.

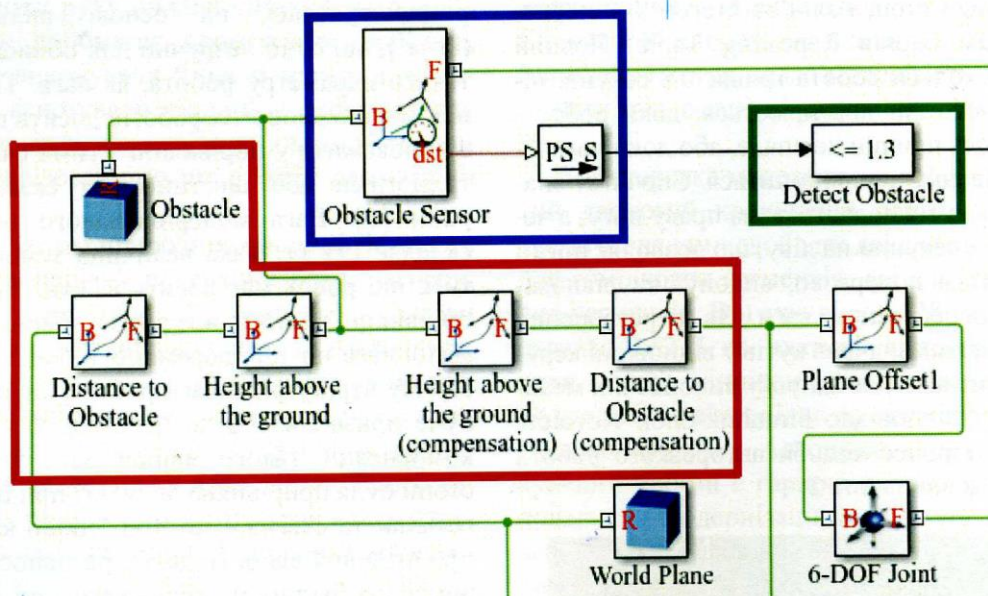


Рис. 3. Ділянка схеми у Simulink, котра відповідає за відображення перешкоди та її виявлення роботом

Після того, як буде виявлена близькість до перешкоди, робот переходить у короткий стан групування і рухається ліворуч, доки не вийде із зони детектування

перешкоди, після чого продовжить рухатися вперед. Логіка обходу роботом перешкод продемонстрована на рис. 4 та рис. 5.

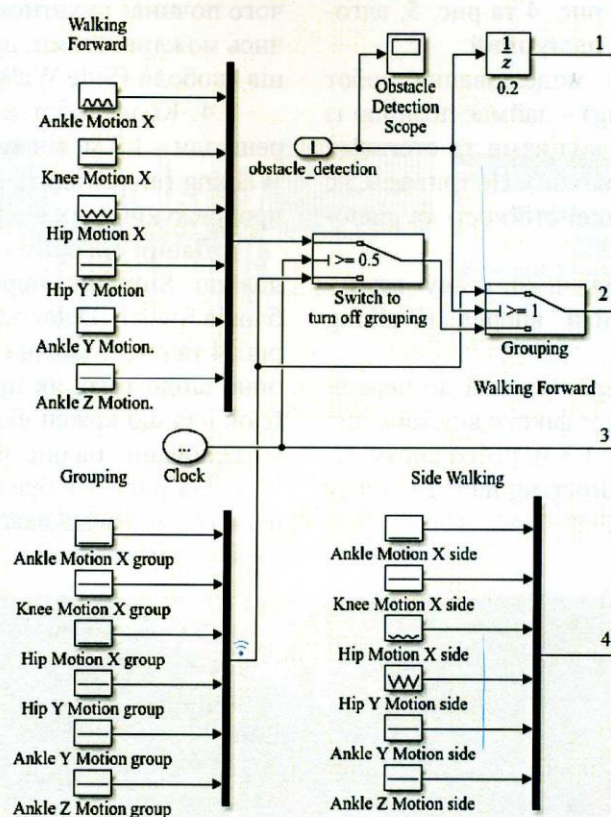


Рис. 4. Ділянка, що відповідає за логіку переміщення робота при виявленні перешкоди

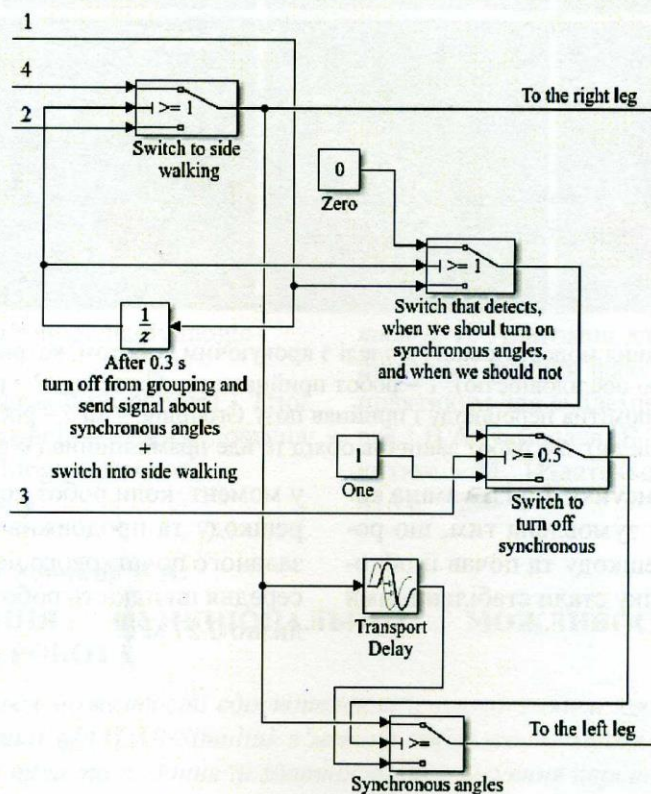


Рис. 5. Продовження ділянки з рис. 4

Відповідно до рис. 4 та рис. 5, алгоритм ходьби робота наступний:

1. На початку моделювання робот групується (Grouping) – займає позицію із зігнутими стопами, колінами та стегнами (-15° , 30° , -30° відповідно). Це триває 0.5 с і необхідно для кращої стійкості та рівноваги робота.

2. Робот виходить зі стану групування і починає йти вперед (Walking Forward).

3. Як тільки прив'язаний до перешкоди Transform Sensor фіксує відстань, що менша або дорівнює 1.3 м, робот знову переходить до стану Grouping на 0.2 с, після

чого починає рухатися ліворуч, користуючись можливостями, що надають 12 степенів свободи (Side Walking).

4. Коли робот покидає діапазон перешкоди – 1.3 м, він виходить зі стану Side Walking і переходить до Walking Forward, продовжуючи рух вперед.

Даний алгоритм реалізовано у середовищі Simulink переважно за рахунок блоків Switch, Delay та Repeating Table. На рис. 4 та рис. 5 можна знайти детальніший опис щодо того, як працює той чи інший блок і за що кожен відповідає. Результати моделювання на рис. 6.

На рис. 7 зображено відпрацювання скрипту, що згадувався у пункті 5 під час переліку нововведень моделі.

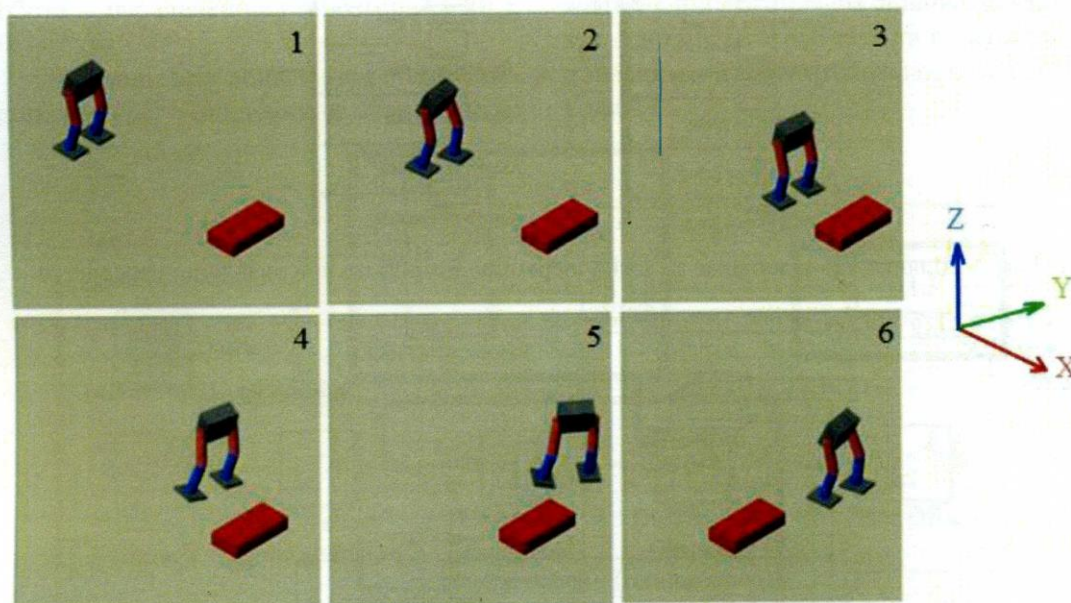


Рис. 6. Відпрацювання модернізованої моделі з крокуючим роботом, котрий обходить перешкоди (з позначеною послідовністю): 1 – робот прийняв позу Grouping, 2 – робот йде (Walking Forward), 3 – робот помітив перешкоду і прийняв позу Grouping, 4 та 5 – робот обходить перешкоду (Side Walking), 6 – робот закінчив обхід та йде прямолінійно (Walking Forward)

Як видно з рисунку 7, різка зміна коливань на графіку зумовлена тим, що робот розпізнав перешкоду та почав її обходити. Покази графіку стали стабільнішими

у момент, коли робот успішно обійшов перешкоду та продовжив рух відповідно до заданого початкового напрямку. При цьому середня швидкість робота становить приблизно 0.27 м/с.

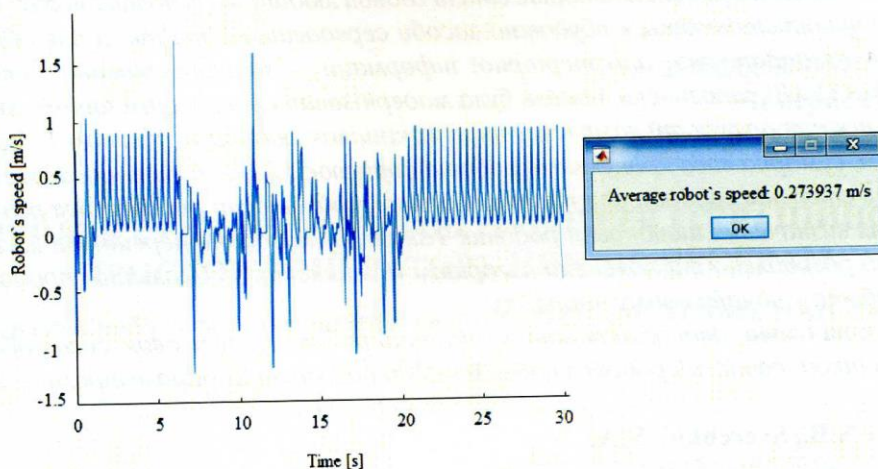


Рис. 7. Графік швидкості робота у кожен момент часу та повідомлення про обчислену середню швидкість робота

Висновки

У статті запропоновано нові функціональні можливості вже існуючої моделі двоногого крокуючого робота. Головні досягнення модернізованої моделі – вміння робота уникати зіткнення з перешкодами, а також володіння вагою та розмірами, що наближені до людських параметрів – мають види на використання робота у небезпечних для людини середовищах з різноманітними перешкодами. Дослідження модернізованої моделі крокуючого робота у MATLAB/Simulink, а також синтезований графік швидкості показали, що робот повністю виконує свою місію.

Література

1. Двунogie ходящие роботы. [Электронный ресурс]. – Режим доступа: <http://robotrends.ru/robopedia/dvunogie-hodyashie-roboty>.
2. Обзор: шагающие роботы | Triol. [Электронный ресурс]. – Режим доступа: <https://triolcorp.ru/blog/post/obzor-shagayushchie-roboty>.

3. Wettergreen D. Robotic walking in natural terrain. Gait planning and behavior-based control for statically-stable walking robots : дис. докт. филос. наук / D. Wettergreen – Pittsburgh PA, 1995. – 134 p.

4. MATLAB and Simulink Robotics Arena: Walking Robot - File Exchange - MATLAB Central. [Электронный ресурс]. – Режим доступа: <https://www.mathworks.com/matlabcentral/fileexchange/64227-matlab-and-simulink-robotics-arena-walking-robot>.

5. Bräunl T. Embedded Robotics: Mobile Robot Design and Application with Embedded Systems, 3rd Ed. / T. Bräunl – New York: Springer Publishing, 2008. – P. 165-184.

6. Замираев В.А., Година Е.З., Никитюк Д. Б. Анатомия для студентов физкультурных вузов и факультетов: учебник и практикум для академического бакалавриата / В.А. Замираев, Е.З. Година, Д.Б. Никитюк. – М.: Издательство Юрайт, 2019. – 416 с.

Репнікова Н.Б., Сверчков В.А.

ВДОСКОНАЛЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ МОДЕЛІ КРОКУЮЧОГО РОБОТА

Розглядаються можливості вдосконалення існуючої моделі крокуючого двоногого робота у середовищі MATLAB/Simulink з метою наближення моделі до реальних параметрів людського тіла та надання їй здатності до уникнення перешкод.

Вдосконалений робот повторює стиль ходьби людини на зігнутих ногах. В ході модернізації використовуються вбудовані засоби середовища Simulink, а для обрахування та виведення додаткової апостеріорної інформації — інтерпретована мова програмування MATLAB. Запозичена модель була модернізована наступним чином: висота робота наближена до висоти комплексу нижніх кінцівок людини зі зростом 175 см і дорівнює 100 см, сумарна вага крокуючого робота дорівнює 172 кг, були впроваджені додаткові шість степенів свободи для надання більшої мобільності роботу, був реалізований скрипт для визначення швидкості робота. Також, була додана переешкода на шляху робота, було розроблено та пояснено алгоритм її уникнення з подальшим продовженням ходьби робота у початковому напрямку.

Ключові слова: двоногий робот, крокуючий робот, модель двоногого робота, уникання переешкод, двоногий робот з наближеними до людських параметрами.

Repnikova N.B., Sverchikov V.A.

IMPROVING THE FUNCTIONAL CAPABILITIES OF THE WALKING ROBOT MODEL

Possibilities of improving the existing model of a walking two-legged robot in the MATLAB/Simulink environment are considered in order to bring the model closer to the real parameters of the human body and give it the ability to avoid obstacles.

The upgraded robot repeats style of human's walking on bent legs. During the modernization, the built-in tools of the Simulink environment are used, and for the calculation and output of additional a posteriori information — the interpreted programming language MATLAB. Existing model was upgraded as follows: height of the robot was approximated to the height of human lower limbs complex with a height of 175 cm and now is 100 cm, total weight of the walking robot is 172 kg, additional six degrees of freedom were introduced to provide greater mobility to the robot, a script was implemented to determine the speed of the robot. Also, an obstacle in the way of the robot was added, an algorithm for obstacle avoidance with the subsequent continuation of the robot's walk in the initial direction was developed and explained.

Keywords: bipedal robot, walking robot, model of two-legged robot, obstacle avoidance, two-legged robot with close to human parameters.